BlackBerry®

Intelligent Security. Everywhere.

Decade of the RATs

Cross-Platform APT Espionage Attacks Targeting Linux, Windows and Android

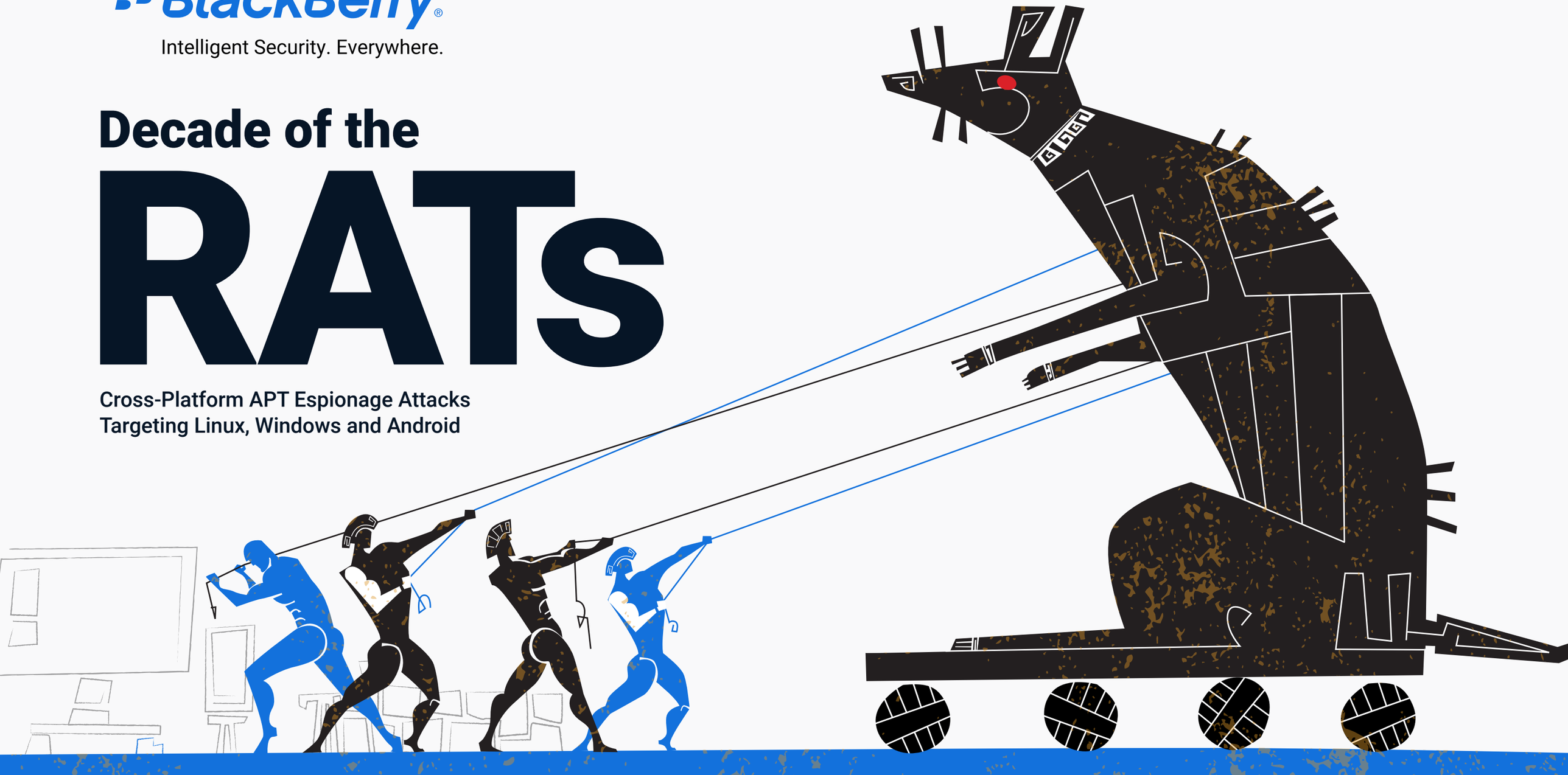

Table of Contents

Executive Summary **3**

Key Findings **5**
- Strategic Intelligence Assessments: 5
- Tactical Intelligence Assessments: 6

Why the Targeting of Linux Systems Matters **7**

The Linux Advantage **8**

Penguin Malware – A Relatively Rare Bird **9**

WINNTI Splinter Cell Targeting Linux **10**

The Linux Splinter Cell Toolset **11**
- PWNLNX1: A Backdoor 12
- PWNLNX1 C2 Infrastructure 14
- The Linux XOR.DDoS Botnet 14
- PWNLNX4: An LKM Rootkit 15
- Linux Build Environments 15
- PWNLNX2: Another Backdoor 17
- PWNLNX2 C2 Infrastructure 17
- PWNLNX6 Updated LKM Rootkit 18
- PWNLNX3: A Backdoor 18
- PWNLNX3 C2 Infrastructure 19
- Introducing WLNXSPLINTER 19
- Lancer – An Installation Script 20
- PWNLNX5 – The Controller 22
- Another Linux Oddity — CASPER Mirai Variant 23

Cellular Division **24**
- PWNDROID4 24
- An Interesting Find 26
- CASPER Goes Mobile – PWNDROID5 27

Windows Base Camp **29**
- Adware? Who Cares? 29

Attribution **36**
- The WINNTI Approach 36

Conclusion **38**
- Legal Disclaimer 38

Appendix **39**
- Linux SHA256 Hashes 39
- WINNTILNX Toolset: 39
- Android SHA256 Hashes 39
- Stolen Code-signing Certificates (2016-2020) and Windows Reference Samples: 39

Works Cited **43**

Published by BlackBerry Limited, 2200 University Ave, E Waterloo, ON Canada N2K 047

Read Blogs.BlackBerry.com, and follow us on Twitter (@BlackBerry) and LinkedIn (https://www.linkedin.com/company/blackberry/)

Executive Summary

The recent Chinese New Year ushered in the *Year of the Rat*, but from the perspective of the many corporations, government agencies and other organizations around the world who continue to be the targets of Advanced Persistent Threat (APT) groups acting in the interest of the Chinese government, recent years could aptly be described as the *Decade of the RATs* - Remote Access Trojans, that is.

As China forges its role as one of the great world powers, it continues to rely upon a blast furnace of cyber espionage operations in order to acquire foreign technologies and intellectual property, to better position itself against the global influence of competing international powers, and to control its own image both at home and abroad.

In response to the pervasive economic espionage threat posed by China, the U.S. Department of Justice (DOJ) announced the China Initiative in November of 2018, a program "focused on preventing and prosecuting thefts of American technology and intellectual property for the benefit of China." Attorney General Jeff Sessions noted "discoveries that took years of work and millions of dollars in investment here in the United States can be stolen by computer hackers or carried out the door by an employee in a matter of minutes…" (Department of Justice, 2018).

In February of 2020, the DOJ organized the *China Initiative Conference* in Washington, DC, where Attorney General William P. Barr stated that the DOJ believes the Chinese government is engaged in a multipronged strategy that includes "cyber intrusions, co-opting private sector insiders through its intelligence services, and using non-traditional collectors, such as graduate students participating in university research projects."

As China forges its role as one of the great world powers, it continues to rely upon a blast furnace of cyber espionage operations in order to acquire foreign technologies and intellectual property.

Barr went on to assert that the DOJ believes these cyber operations are tied to the Chinese government. He said, “With respect to remote computer intrusions, for example, the [DOJ] indictment of APT 10 hackers in December 2018 outlined a global campaign, associated with the Chinese Ministry of State Security, targeting intellectual property and confidential business and technology information…” (Department of Justice, 2020).

While Chinese IP theft is now a story old enough for the history books, there continue to be new chapters to add with new lessons to learn for security teams and the organizations they serve.

In this report, BlackBerry researchers examine the activities of five related adversarial groups who have spent the better part of the last decade successfully targeting organizations in cross-platform attacks while operating relatively, if not entirely undetected in multiple strategic and economic espionage operations.

The report details how this quintet of threat actor groups have been focused on an often-overlooked platform: Linux® servers that comprise the backbone of the majority of large data centers responsible for the some of the most sensitive enterprise network operations. And it further reveals the link between a previously unidentified Linux malware toolset and one of the largest Linux botnets ever discovered.

The newly discovered Linux malware toolset included two kernel-level rootkits that rendered executables extremely difficult to detect, making it highly probable that the number of impacted organizations is significant and the duration of the infections lengthy. This report provides analysis of the attacks, the toolset, the rootkits, the other malware, and the infrastructure involved.

The research also provides analysis of attacks designed to elude defenders through the use of Windows® malware that uses adware code-signing certificates, a tactic that the attackers hope will increase infection rates as any red flags are dismissed as just another blip in a constant stream of adware alerts. This report examines multiple samples of malware accompanied by the adware code-signing certificates.

The researchers also look at the targeting of another often-neglected attack vector: the mobile devices that increasingly make up a significant portion the enterprise network perimeter. A previous report from BlackBerry® researchers, titled *Mobile Malware and APT Espionage: Prolific, Pervasive, and Cross-Platform* (BlackBerry, 2019), examined how APT groups have been leveraging mobile malware in combination with traditional desktop malware in ongoing cross-platform surveillance and espionage campaigns. This report continues the analysis of this trend with an examination of some newly identified Android™ malware.

The report also delves into the curious case of a mobile remote access trojan (RAT) that was developed by an APT group nearly two years prior to the commercial availability of a popular remote administration penetration testing tool that has strikingly similar code structure and characteristics, raising questions about the origins of each.

This report provides a threat intelligence assessment of the strategic and tactical use of novel malware and attack techniques employed by several threat actors. The conclusions drawn here represent the best judgments of the researchers based on data examined.

While Chinese IP theft is now a story old enough for the history books, there continue to be new chapters to add with new lessons to learn for security teams and the organizations they serve.

Key Findings

Strategic Intelligence Assessments:

- **Targeting Linux:** Adversaries assessed to be acting in the interests of the Chinese government have strategically targeted Linux servers for years precisely because the Linux operating system is not typically a primary focus of security solutions. Defensive coverage within Linux environments is immature at best, and robust endpoint protection (EPP) and endpoint detection and response (EDR) products are often inadequately utilized or lack the capabilities to defend them. It was assessed that the groups examined in this report are using Linux servers as a “network beachhead” for other operations – that is, as a highly available attack vector that is always-on and poorly defended.
- **APT Groups Coordinating:** Persistent threats rarely operate in a single domain, and the five groups assessed to be related to the APT originally identified as WINNTI GROUP in previously published research are no exception. Many of the techniques used in one operating environment have been readily translated for use in others. Cross-platform and open-source tools are more readily available now than ever, and the APT groups examined in this report have already exploited this fact.
- **Objective Blending and Overlap:** BlackBerry researchers observed the continued blending of financially motivated and targeted espionage activity by the five groups under examination in this report. The more traditional criminal approaches to network exploitation are equally effective in their intelligence gathering as they are in generating revenue. Attacks that look like dragnet, “spray and pray” efforts can also yield targeted reconnaissance intelligence for other operations, and strategic platform and supply-chain compromises are becoming increasingly commonplace.
- **Attackers for Hire:** It is assessed with high confidence that the APT groups examined in this report are likely comprised of civilian contractors working in the interest of the Chinese government who readily share tools, techniques, infrastructure, and targeting information with one another and their government counterparts. This reflects a highly agile government/contractor ecosystem with few of the bureaucratic or legal hurdles that can be observed in Western nations with similar capabilities and provides a level of plausible deniability for the Chinese government.

These groups target Red Hat Enterprise, CentOS, and Ubuntu Linux environments systemically across a wide array of industry verticals for the purposes of espionage and intellectual property theft.

Tactical Intelligence Assessments:

- **The WINNTI Approach:** Five APT groups acting in the interest of the Chinese government and assessed to be employing WINNTI-style tooling have taken strategic aim at Linux servers that serve a critical role in enterprise network environments and have done so while remaining relatively undetected for nearly a decade. These groups target Red Hat Enterprise, CentOS, and Ubuntu Linux environments systemically across a wide array of industry verticals for the purposes of espionage and intellectual property theft. The APT groups examined include the original WINNTI GROUP, PASSCV, BRONZE UNION, CASPER (LEAD), and a newly identified group BlackBerry researchers are tracking as WLNXSPLINTER. All five groups are assessed to be related given the distinct similarities in their tools, tactics and procedures (TTPs) employed and referred to in this report as the WINNTI approach.
- **The Linux Connection:** The APT groups examined in this report have traditionally pursued different objectives and focused on a wide array of targets. However, it was observed that there is a significant degree of coordination between these groups, particularly where targeting of Linux platforms is concerned, and it is assessed that any organization with a large Linux distribution should not assume they are outside of the target sets for any of these groups.
- **The XOR DDoS Botnet Connection:** It was also observed that the malware used by WINNTI GROUP very closely resembles that used in the massive Linux XOR DDoS botnet first identified in September of 2014, to the extent that BlackBerry researchers have judged the botnet to have been a tool developed by this group.
- **Code Similarities:** A PASSCV Android implant examined in this report very closely resembles code marketed as the penetration testing tool NetWire for Android, yet the malware is shown to have been compiled nearly two years before the commercial NetWire tool was first made available for purchase.
- **Hiding in Plain Sight:** The APT groups examined in this report have shifted from signing malware certificates stolen from video game companies to signing malware with certificates stolen from adware vendors, resulting in very low detection rates. It is assessed that this was being done to bypass network defenders by hiding malware within the high volume of innocuous adware alerts large organizations typically receive in any given day.
- **Cloud Migration:** It has been observed that there has been a shift in infrastructure hosting towards the more frequent adoption of established, legitimate cloud services, presenting a challenge to defenders' assumptions regarding the monitoring of trusted network traffic within their organizations' networks.

Why the Targeting of Linux Systems Matters

Linux is arguably the most important yet least user-friendly operating system in the world. While most people are unlikely to be using it on their desktop at work or at home, Linux dominates the backend infrastructure of large modern data centers.

Linux runs the stock exchanges in New York, London and Tokyo, and nearly all the big tech and e-commerce giants are dependent on it, including the likes of Google, Yahoo, and Amazon. Most U.S. government agencies and the Department of Defense also rely heavily on the Linux operating system, and it runs virtually all of the top one-million websites and 75% of all web servers (Netcraft, 2019). Linux powers 98% of the world's most advanced supercomputers, and if you or your organization stores data in the cloud, you'll find Linux running in the background more than 75% of the time (Linux Foundation, 2020).

Given the open-source nature of Linux, it is generally considered to be a more secure and require less maintenance, making it the ideal operating system for backend servers. Behind the scenes at government agencies, universities, and corporations around the world, you'll find Linux on servers that house sensitive data as well as those that keep critical systems up and running.

Linux keeps the lights on when the employees have all gone home for the night, which means servers running Linux are trusted to be always-on and always accessible. These qualities have made Linux the operating system of choice for many systems administrators – and also a strategic target for state-sponsored espionage operations.

In this section of the report, BlackBerry researchers lay bare how a quintet of APT groups acting in the interest of the Chinese government - assessed to be offshoots of the original WINNTI GROUP - developed the capability to exploit the "always-on, always available" nature of Linux servers to establish an operations beachhead in targeted networks while remaining almost entirely undetected for nearly a decade.

BlackBerry researchers lay bare how a quintet of APT groups acting in the interest of the Chinese government - assessed to be offshoots of the original WINNTI GROUP - developed the capability to exploit the "always-on, always available" nature of Linux servers to establish an operations beachhead in targeted networks while remaining almost entirely undetected for nearly a decade.

The Linux Advantage

Linux servers, whether located on-premises or with a cloud provider, are an ideal and strategic target of espionage for several reasons:

- Compromising Linux web servers allows for the exfiltration of massive amounts of data that can be obscured within the high volume of daily web traffic
- Compromising Linux database servers provides attackers a greater chance of finding valuable data like sensitive intellectual property, trade secrets, or lists of employee usernames and passwords relatively quickly
- Compromising Linux jump-boxes, aka bastion or proxy servers, erases a layer of protection typically relied upon by most corporate networks to separate internal networks from external threats

All three types of servers described above – web, database, and proxy – are designed to be "up" all the time, meaning the same benefits they provide system administrators (continuous, reliable network access) are also afforded to the attackers who compromise them, making them a perfect staging area from which to penetrate other areas of the network.

What's more, all the source code for the Linux distributions commonly seen in corporate and government environments, including Red Hat Enterprise, Ubuntu, and CentOS, is freely available to examine. This plays to one of an APTs key strengths: it allows knowledge of the operating system to be more readily exploited and for the tools designed to circumvent security to be more effective.

As described in *Chinese Industrial Espionage* (Hannas, Mulvenon, & Puglisi, 2013), the Chinese are more adept than any other nation in absorbing, translating, and leveraging open-source material: "Not only does China invest far more effort in open-source collection than other countries, the 'back-end' components – analysis, customer interaction, and feedback to collectors – also play a much larger part, as befits a nation whose progress depends more on adaptation than innovation."

In the attacks BlackBerry observed, the open Linux platform has enabled Chinese actors to develop backdoors, kernel rootkits, and online-build environments at a high level of complexity and specificity, with the end result being a toolset specifically designed to be harder to detect. Compounding low detection rates inherent in the malware design is the relative lack of coverage quality and features in malware detection solutions for Linux available on the market today.

Linux's command-line interface also makes it less widely accessible, which means it is usually administered by a smaller number of skilled systems administrators. In contrast, practically everyone from the corner office to the mailroom uses desktop computers running either Windows or MacOS®, so most security companies have focused more of their research and development on products for the front office as opposed to the server rack.

The combination of poor security solution coverage for Linux and highly tailored, complex malware has resulted in a suite of adversary tools that has largely - if not entirely - gone undetected for years.

Penguin Malware – A Relatively Rare Bird

Before proceeding with a discussion of the findings, readers of this report may find it helpful to know something of the Linux threat landscape more generally for context:

Groups associated with the state or state-sponsored efforts of at least three governments have been found to develop and deploy Linux malware: China, Russia, and the United States. A class of Linux malware called Derusbi has been known to be used by APT groups acting in the interest China, including LEVIATHAN (APT40), DEEP PANDA, AXIOM, and APT41 (MITRE, 2017). In 2014, Kaspersky discovered that the Russian group TURLA was also deploying Linux malware (Baumgartner & Raiu, 2014), and another group Kaspersky identifies as THE EQUATION GROUP (generally thought to be the NSA) has also targeted Linux servers extensively.

In May of 2019, researchers at Chronicle detailed several Linux implants believed to be associated with APT41 based on an examination of a network protocol similar to the DoubleDoor implants (Chronicle, 2019). The protocol used a single-byte incrementing XOR key for string obfuscation and employed an LKM rootkit based on the open-source "Azazel" project.

However, in comparison to the volume of malware directed at Windows and MacOS operating systems, Linux malware is observed and written about much less often. This is reflective of its relatively low rate of detection and relatively low frequency of being encountered in incident response engagements.

The paucity of public knowledge about the Linux threat landscape presents obvious challenges in piecing together attack scenarios and understanding them completely. But at least one assessment can be made with near complete certainty: the amount and age of the Linux malware tools wielded by the threat groups discussed in this report are confirmation that the targeting of Linux has been wildly successful.

But at least one assessment can be made with near complete certainty: the amount and age of the Linux malware tools wielded by the threat groups discussed in this report are confirmation that the targeting of Linux has been wildly successful.

WINNTI Splinter Cell Targeting Linux

BlackBerry researchers have assessed that there are at least five APT groups acting in the interest of the Chinese government which together comprise a “splinter cell” that targets enterprise Linux distributions, all of which are related to one another and to an APT identified in earlier research as WINNNTI GROUP.

For the first time, BlackBerry researchers have assessed that these groups are all sharing a previously unidentified Linux malware toolset referred to in this report as the WINNTILNX toolset. It should be noted that these groups have also been observed targeting other platforms as well, including Windows, Android, and MacOS.

Four of these five groups are already known to the security community as PASSCV, BRONZE UNION (aka APT27, EMISSARY PANDA), a group tracked internally as CASPER (aka LEAD), and the original WINNTI GROUP. But the fifth Linux splinter cell group, which BlackBerry researchers are tracking as WLNXSPLINTER, is discussed for the first time in this report. These threat actor groups share three important characteristics:

- All five groups examined in this report have been observed attacking video game companies to steal code-signing certificates which they used to sign their malware, as well as attacking the gaming companies for criminal purposes to produce revenue.
- All five groups share tools, suggesting several possible scenarios: a formal “digital quartermaster” arrangement (a la FireEye); an informal “hacker forum” type of tool-swap; personnel overlap between the groups; or a re-tasking of the same groups toward different target sets.
- Their targets run the gamut of nearly all verticals, and activities range from simple cybercrime to full-blown economic espionage, and from internal monitoring of politically dissenting populations to more traditional military and strategic nation-state espionage. These groups’ collective palette is wide and well-developed, touching nearly every industry sector across a huge geographic area.

At least one of these groups, referred to in this report as Kaspersky’s original WINNTI GROUP, can now be linked more strongly - if not explicitly - to China’s Ministry of State Security (MSS) based on this research, a discussion taken up in earnest in the *Attribution* section near the end of this report.

These groups’ collective palette is wide and well-developed, touching nearly every industry sector across a huge geographic area.

The Linux Splinter Cell Toolset

BlackBerry researchers have discovered that the Linux splinter groups have developed and deployed the following tools, collectively referred to as the WINNTILNX toolset. Included in the toolset are:

- Three backdoors, all of which are unique variants and designated as:
  - PWNLNX1
  - PWNLNX2
  - PWNLNX3
- Two rootkits which are deployed simultaneously with the backdoors, all of which are the Linux Kernel Module variety and designated as:
  - PWNLNX4
  - PWNLNX6
- Two build-groups which are used to construct the rootkits on-the-fly and tailor them to their targets, designated as:
  - Group 1 (online)
  - Group 2 through Group 6 (local)
- An installer script used to remotely compile, download, and install both an LKM rootkit and a backdoor on the target, designated as:
  - Lancer
- A Control Panel used by the attackers to run the command-and-control (C2) infrastructure and issue commands to the rest of the malware suite (for both Windows and Linux) and designated as:
  - PWNLNX5
- A massive Linux botnet:
  - XOR.DDoS which was first identified in September of 2014 and known to have been used to attack video game companies in Asia, among others

When BlackBerry researchers first uncovered this malware suite, they were curious to know how long it had been in use but determining that proved to be a nontrivial task. Linux malware executables are referred to by the term ELF, which stands for *Executable and Linkable Format*. Unlike their Windows counterparts (called PEs or *Portable Executables*), ELFs do not possess a compiler time/date stamp, which makes it difficult to discern exactly when Linux samples were created.

However, ELFs often contain a reference to the compiler used to create them, and the age of that compiler can provide a very rough indication of the age of the ELF. Here's how this was accomplished:

The simple command line "objdump -s --section .comment *{full path to binary}*" can be used to extract this information from an ELF file if it is present. ELF files may also contain references to the source files they were compiled from, in this case:

- crtstuff.c
- down.c
- encrypt.c
- file.c
- hide.c
- main.c
- portforward.c
- portmap.c
- pty.c
- socket.c
- udp.c
- up.c
- crc32.c
- dns.c

The earliest sample BlackBerry researchers identified using this method was compiled with "GCC: (GNU) 4.4.7 20120313 (Red Hat 4.4.7-3)" which was released on March 13, 2012, suggesting that WLNXSPLINTER has potentially been in use for roughly the last eight years.

The age of this suite is a salient point and should not be underestimated. Though there was not enough data from incident response engagements to paint a complete picture of the attack chain, its longevity combined with the low (in some cases zero) detection rates for the malware suggests that the suite has been successful in establishing and maintaining itself in target environments for quite some time.

It's also important to note that the backdoors communicated both to internal as well as external IP addresses. This indicates that the groups attacked servers that were both deliberately segmented to keep them from connecting to the internet (a practice often judged to make them more secure), as well as connecting to web servers that reached outside the target organization.

The infection of internal-only servers indicates that the attackers were either successful in exploiting "crown jewel-type" data normally kept in such vaults, or that they had planned ahead and established a backup point of access in case other avenues were discovered and blocked.

PWNLNX1: A Backdoor

The WINNTI GROUP made very few alterations to the backdoor designated as PWNLNX1 over the years, with the exception of some minor feature additions. Yet even after all this time, the majority of the samples examined have a zero-detection rate in the industry's most commonly used virus repository.

PWNLNX1 was designed to work with a Local Kernel Module (LKM), which enabled it to perform a number of rootkit functions like bypassing iptables, hiding files, hiding processes, hiding threads, and hiding network connections. It also provided the attackers with the ability to upload and download files, enumerate and manipulate files and directories, access an interactive shell, forward traffic and ports, and modify and connect to the embedded command-and-control (C2) servers over TCP and UDP, as well as over IPv4 and IPv6.

The backdoor encoded its network callback information using a simple operation against XOR keys "CB2FA36AAA9541F0" or "BB2FA36AAA9541F0" was only observed in earlier samples.

It should be noted that the group we call CASPER (aka LEAD) utilized another unique XOR key in its version of PWNLNX1: "1A2FB36DAC95E1F9". The corresponding C2 domains were resolved using external servers hard coded into the files, either "8.8.8[.]8" or "114.114.114[.]114."

An example of the encrypted and decrypted configurations are presented below, along with the simple python snippet to perform this operation without preserving the null values:

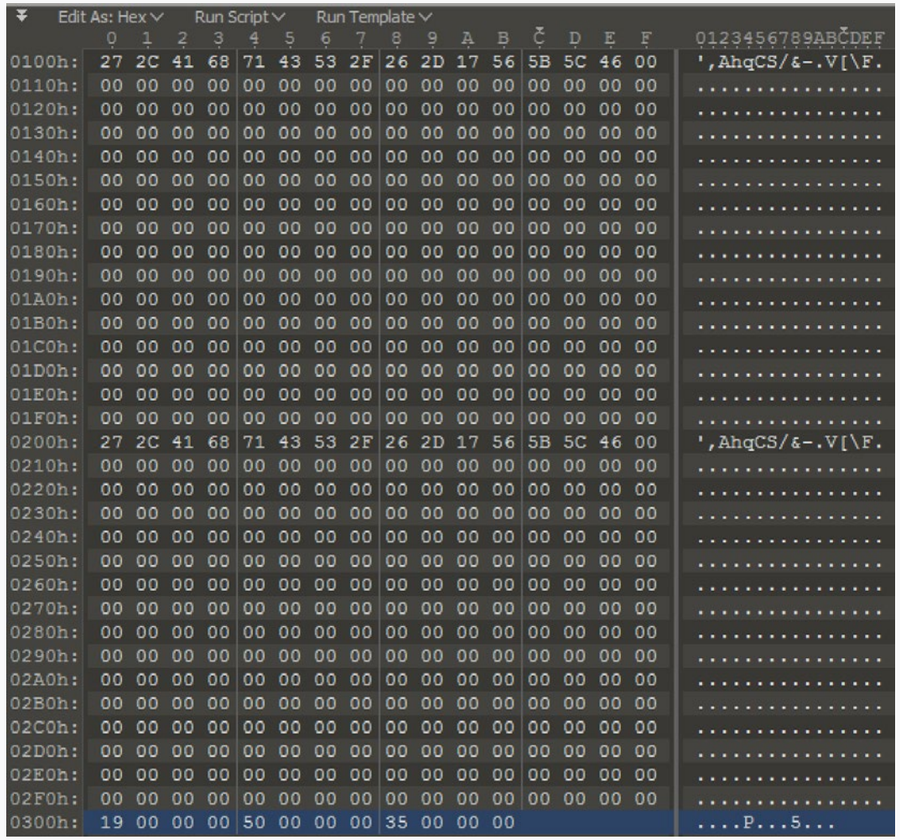

Figure 1: Encoded Configuration of PWNLNX1

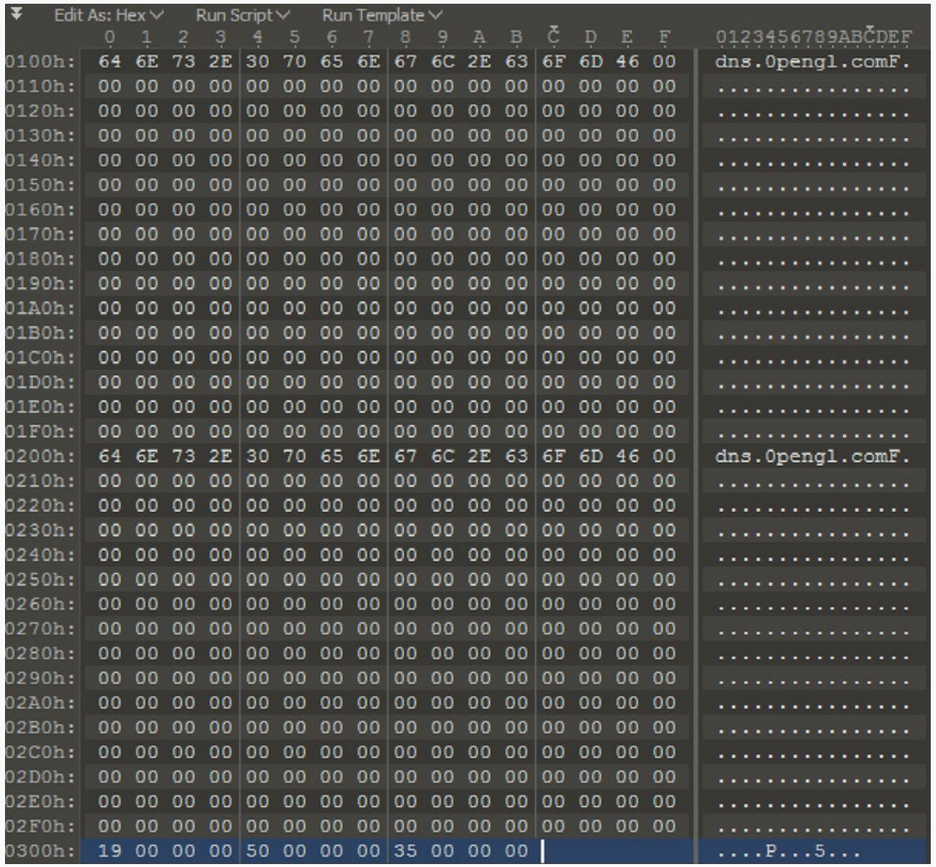

Figure 2: Decoded Configuration of PWNLNX1

```
def rolling_xor(buf, key):
   out = ‘’
   k = 0
   for i in buf:
      if k == len(key):
         k = 0
      out += chr(ord(i) ^ ord(key[k]))
      k += 1
   return out
rolling_xor(config_block, ‘CB2FA36AAA9541F0’)
```

Figure 3: Python Pseudocode to Decrypt Configuration Block of PWNLNX1

BlackBerry researchers were able to link the use of this backdoor to multiple intrusion sets that have been previously publicly identified. They did this based upon the observations of the distinct C2 infrastructure, subdomain similarities, and other unique characteristics.

Given the diverse range of targets for each group in the Linux splinter cell quintet, it is assessed with high confidence that there was likely some sort of common direction, nexus, or at the very least shared tooling between the groups.

Given the diverse range of targets for each group in the Linux splinter cell quintet, it is assessed with high confidence that there was likely some sort of common direction, nexus, or at the very least shared tooling between the groups.

PWNLNX1 C2 Infrastructure

WINNTI GROUP:	PASSCV:	CASPER (LEAD):
ark.aeriagames[.]us	dns.0pengl[.]com	bot2.googlerenewals[.]net
www.alidnx[.]com	35.201.147[.]249	linux2.googlerenewals[.]net
10.79.120[.]10		sdfaswaed2.nokiadns[.]com
10.79.250[.]80		serconsole.vicp[.]cc
10.79.4[.]131		
mi.btmods[.]net (maybe BRONZE UNION)		
us.btmods[.]net (maybe BRONZE UNION)		

BRONZE UNION (APT27/EMISSARY PANDA):
y3dx36f6.love6d[.]com
kdwontyraqdswlqm[.]ossrescue.com
l3wpk9kmumodtkr8[.]ibmassist.com

The Linux XOR.DDoS Botnet

While BlackBerry researchers were unable to recover the local kernel module from an active infection, examination of how WINNTI GROUP's PWNLNX1 interacted with its rootkit provided a startling discovery: an explicit link to one of the largest Linux botnets ever, which was discovered in September of 2014, and dubbed "Linux.XorDDoS" (Malware Must Die!, 2014).

The Xor.DDoS botnet earned notoriety in 2015 for its high-bandwidth attacks of up to 150 GBPS. According to researchers at Akamai, the botnet was observed attacking 20 targets per day, 90% of them in Asia, with video game companies leading the target list (Akamai, 2015).

Akamai observed that the botnet grew in size after using brute force attacks to obtain the password for the target Linux server. The attackers then simply logged in to the server to drop the botnet malware (Akamai, 2015). Curiously, despite its size and relative firepower, news of widespread denial of service reports has proven difficult to find. This begs the question: Why build a giant DDoS botnet if you don't intend to disrupt websites?

In investigating further, BlackBerry researchers found that PWNLNX1 utilized a device named "/proc/rs_dev" for rootkit functionality. This is the exact same device name used in the botnet. What's more, early PWNLNX1 samples even used the exact same string as an XOR key for network traffic obfuscation: "BB2FA36AAA9541F0".

These combination of these factors – the age of the malware, repeated use of the exact same XOR key, code reuse, and targeting of the video game industry - indicated either there was a sharing between the WINNTI GROUP and the group behind the botnet or they were in fact one and the same. Of those two possibilities, BlackBerry researchers judged the latter to be more likely, and that WINNTI GROUP was behind the Xor.DDoS botnet.

This was a surprising find, as it was not expected to find the kernel modules used by PWNLNX1 had employed different ioctl codes for the same functionality. This told us that the Linux Kernel Modules (LKMs) were different despite using many of the same function names.

A little more digging lead us to conclude that both were instead based off of the open-source "Suterusu Rootkit" (Coppola, Suterusu Rootkit: Inline Kernel Function Hooking on x86 and ARM, 2013). This, in contrast, wasn't at all surprising given China's aforementioned affinity for and skill in leveraging open-source material.

Avast analyzed the Linux.Xorddos rootkit in detail in a blog post from January of 2015 (Avast Threat Intelligence Team, 2015). Of particular note was for any LKM to function properly, its "vermagic" value must match that of the currently installed kernel headers on the victim's system.

This meant that each rootkit would have to be specifically compiled for the victim system it was deployed on. The Linux.Xorddos rootkit had a solution for this problem through an online build server that was accessed through a series of HTTP GET requests, as detailed in Avast's blog. For the sake of brevity, this report will simply point out the second GET request because it turned out to be of particular interest. Indeed, it factored into other portions of this research, as described below.

Let's look at this second GET request in more detail here. Below you'll find an attempt to start to break it down, beginning with the request to "/compiler?". Note the first three steps taken were lookup hash of kernel, enter username to build server, enter password:

Additional Parameter	Value	Function
iid=	CE74BF62ACFE944B2167248DD0674977	Lookup Hash of Kernel
username=	admin	Username to Access Build Server
&password=	admin	Password to Access Build Server
ip=	103.25.9[.]245:8005|103.240.141[.]50:8005[snip]	C2 Servers
&ver=	3.8.0-19-generic\ SMP\ mod_unload [snip]	Full Kernel Version
kernel=	3.8.0	Base Kernel Version

Figure 4: Breakdown of Build Server Request (We'll Come Back to This…)

PWNLNX4: An LKM Rootkit

After further analyzing how PWNLNX1 interacted with its rootkit, BlackBerry was able to recover several different iterations of the PWNLNX4 rootkit. At the time of writing this report, most of the identified rootkits were undetected by any antivirus vendor. The rootkits were modified versions of the "Suterusu" rootkit; each contained modifications to directly patch the TCP and UDP socket tables, process tables, and file tables.

In still another example of skillful exploitation of open-source material, the code responsible for these modifications appears to have been directly lifted from a book written by Ivan Sklyarov. Note how all the function names and code are identical to those described in *Programming Linux Hacker Tools Uncovered: Exploits, Backdoors, Scanners, Sniffers, Brute-Forcers, Rootkits* (Sklyarov, 2007).

Linux Build Environments

So, what else can be gleaned from a more or less open-source rootkit? BlackBerry researchers identified several different groups of build environments based upon leftover path information:

Build Group1:

- **Build Environment 1:** "/opt/uOnlineBuilder64/core/build/yang/rk"
  - `/opt/uOnlineBuilder64/core/build/yang/rk/lkm.c`
  - `/opt/uOnlineBuilder64/core/build/yang/rk/autoipv6.mod.c`
  - `“”/build/yang/AB1167FF11C7B8642D547D84AEDD8B46/2.6.32-358.el6.x86_64`

- **Build Environment 2:** /opt/uOnlineBuilder64/core/build/hehe/rk
  - `/opt/uOnlineBuilder64/core/build/hehe/rk/lkm.c`
  - `/opt/uOnlineBuilder64/core/build/hehe/rk/autoipv6.mod.c`
  - `“”/build/hehe/4F666C7AA5F592EF64E9B2AFFE267B0F/2.6.32-754.6.3.el6.x86_64`

- **Build Environment 3**: /opt/uOnlineBuilder64/core/build/maomao/rk
  - `/opt/uOnlineBuilder64/core/build/maomao/rk/lkm.c`
  - `/opt/uOnlineBuilder64/core/build/maomao/rk/ip4tables.mod.c`
  - `“”/build/maomao/01944A09FD7592DDFEF4AD4825AB6329/2.6.32-431.11.29.el6.ucloud.x86_64`

Build Group 2:

- Build Environment: /root/Desktop/dns
  - `/root/Desktop/dns/lkm.c`
  - `/root/Desktop/dns/snd_raw.mod.c`
  - `/usr/src/kernels/2.6.32-642.el6.x86_64`
  - `/usr/src/kernels/2.6.32-431.el6.x86_64`

Build Group 3:

- Build Environment: /var/tmp/.1
  - `/var/tmp/.1/lkm.c`
  - `/var/tmp/.1/autoipv6.mod.c`
  - `/usr/src/kernels/3.10.0-693.2.2.el7.x86_64`

Build Group 4:

- Build Environment: /var/tmp/Linux_Server
  - `/var/tmp/Linux_Server/lkm.c`
  - `/var/tmp/Linux_Server/dhcp.mod.c`
  - `/usr/src/kernels/2.6.32-358.14.1.el6.x86_64`

Build Group 5:

- Build Environment: /dev/shm/2.6.32microcode
  - `/dev/shm/2.6.32microcode/lkm.c`
  - `/dev/shm/2.6.32microcode/microcode.mod.c`
  - `/usr/src/kernels/2.6.32-358.14.1.el6.x86_64`

Build Group 6:

- Build Environment: //home/rhudgins/2.6.32floppy
  - `/home/rhudgins/2.6.32floppy/lkm.c`
  - `/home/rhudgins/2.6.32floppy/ipmi_devintf.mod.c`
  - `/usr/src/kernels/2.6.32-358.14.1.el6.x86_64`

Based upon included path information, Groups 2-6 were likely compiled directly on victim machines, not online. In each case the attacker had already obtained access to the server, e.g. through compromised credentials. Group 1 grabbed our interest because some of the additional path information indicated that an online build environment existed which could potentially compile and deliver the rootkits on-the-fly based upon the version of the kernel headers which it tracked, and not just by an MD5 hash but by username as well.

The usernames seen in the Group 1 path names above weren’t terribly revealing but were interesting to note nonetheless because they included: “yang”, “hehe”, and “maomao”. Here’s where the “username” and “password” asked for in Figure 2 above plays in: in order to begin accessing the online build server, a username and password must be provided, and here’s where “yang” and crew signed in before getting down to business.

BlackBerry researchers found that each of the victims’ kernel versions indicated they were all running various versions of Red Hat Enterprise Linux or CentOS – by using this information it was possible to discover the earliest possible compromise date for each victim (Red Hat, 2019).

Kernel Version	Release Date
3.10.0-693.2.2.el7.x86_64	July 31, 2017
2.6.32-754.6.3.el6.x86_64	June 19, 2018
2.6.32-642.el6.x86_64	May 10, 2016
2.6.32-431.el6.x86_64	November 20, 2013
2.6.32-358.el6.x86_64	February 21, 2013

Figure 5: Kernel Versions Indicating Earliest Possible Compromise Dates

PWNLNX2: Another Backdoor

BlackBerry researchers discovered a second variant designated as PWNLNX2 which first appeared around 2017 and was used into early 2018. BlackBerry researchers ascribed these particular variants to the threat groups previously identified as PASSCV and BRONZE UNION (aka APT27, EMISSARY PANDA).

These backdoors were extremely similar to the earlier PWNLNX1 samples and contained the same function names, backdoor functionality, LKM rootkit name and ioctl's, commands from the C2, and they were even compiled from source files bearing the exact same names.

However, the backdoors were significantly larger in size, weighing in at nearly one megabyte. As with earlier samples discussed, they were all undetected in the common malware repository with the exception of one sample, which was mistakenly identified by a single vendor as belonging to a DDoS botnet. BlackBerry researchers originally identified these files from the unique XML output of a function responsible for conducting file-based reconnaissance of the victim's machine and then used this to locate both a 32-bit and 64-bit versions.

Additional functions were present which would determine information about the current operating system and kernel version from the file "/proc/sys/kernel/osrelease". If the version was less than or equal to 2.6.11, the backdoor would terminate. The backdoors also contained some advanced functionality that would enable them to enumerate and manipulate pages of physical memory – features that were not present in earlier versions.

PWNLNX2 C2 Infrastructure

PASSCV:	BRONZE UNION / APT 27:
dns.0pengl[.]com	tab.dellrescue[.]com
linux.cocoss2d[.]com	
linux.css2[.]com	
linux.unitys3d[.]com	

Three of the above domains were previously identified in October of 2016 as belonging to the PASSCV group by BlackBerry (Cylance Threat Research Team, 2016); however, no associated Linux malware samples were identified at the time. These domains currently resolve to IP addresses that reside within several large cloud providers' infrastructures. This tactical shift makes complete sense as cloud infrastructure provides a cost-effective solution that's easily managed and deployed. As an added benefit, network defenders tend to trust IP ranges that belong to well-known cloud provider companies.

PWNLNX6 Updated LKM Rootkit

Based upon unique submitter identification numbers, BlackBerry researchers were able to locate another modified version of the Suterusu Rootkit which the attackers referred to “xinted.ko”. This particular version of the rootkit, which is designated here as PWNLNX6, was compiled using a newer version of GCC (GNU Compiler Collection) with an exact build command matching that of the one used for “Build Group 3” mentioned above. It looks like this:

- Build Environment: /tmp/suterusu
  - `/tmp/suterusu/main.c`
  - `/tmp/suterusu/util.c`
  - `/tmp/suterusu/module.c`
  - `/tmp/suterusu/xinted.mod.c`
  - `/usr/src/kernels/3.10.0-693.17.1.el7.x86_64`

Several functions were absent, notably the routines to directly patch the TCP and UDP tables. However, the most significant change was the creation and implementation of a custom Netlink Protocol to replace the previously used ioctl codes. The following blog gives a good high-level overview of the Netlink Protocol and how to implement a custom one: *Implementing a New Custom Netlink Family Protocol* (Jang, 2019).

In essence, this change enabled the attackers to communicate more efficiently from the kernel to user side of the target machine. A different Netlink protocol appears to have also been implemented within the original Suterusu source code around the same time in June of 2017, one which may have provided the operators inspiration for their own protocol. In regular English: all of these changes meant there was at least one more variant in this family of backdoors that BlackBerry researchers had yet to identify.

PWNLNX3: A Backdoor

BlackBerry researchers went digging a little further, and while the implant that implemented the newer Netlink Protocol was not located, yet another 2018 variant within the WINNTILNX toolset turned up, designated here as PWNLNX3.

The PWNLNX3 samples were significantly larger still, weighing in at nearly four megabytes. Surprisingly, three of these samples were detected by approximately twenty different vendors, give or take, under various monikers. “Linux.Agent.by” was perhaps the most accurate industry detection, but another earlier sample went fully undetected as late as February 10, 2018.

Two of the identified samples referenced a new rootkit module named “/proc/policy4_dev”. They similarly utilized a different ioctl code that was one-byte off from previously identified samples, “0x46375828”, to interact with the rootkit.

The code used to update the backdoor to a newer version was reworked, but otherwise the core functionality of the backdoor was more or less unchanged from earlier samples. Two new functions implemented named “HandleUpdate” and “execUpdate” which would download a file from the C2 server using the command line “wget -P” to the directory “/tmp” and run “chmod 777” on the file, making it world readable, writable, and executable before running the update package. Several newer and distinct C2 servers were used to administer these particular samples.

PWNLNX3 C2 Infrastructure

PASSCV:	WLNXSPLINTER:
b.zabbixx[.]com	cachecdn.moegoo[.]com
gs.gw688[.]org	
orabbix.zabbixmonitor[.]net	
yum.anydesk[.]me	
yum.nortonvirus[.]org	
zabbix.symanteprotection[.]com	

The domains “zabbixx[.]com”, “gw688[.]org”, and “zabbixmonitor[.]net” were all first registered on June 19, 2018 using “dns.com”, then transferred to “1-api.org” on June 29, 2019, and then transferred back to “dns.com” on July 5, 2019. Using this particular registration pattern, BlackBerry researchers identified that the following domains are also highly likely to be under the attacker’s control: live800kf[.]com, observeit[.]org, shterm[.]net, vncviewer[.]org.

BlackBerry researchers attribute the above collection of domains to the PASSCV group with moderate to high confidence based upon other subdomains resolving to a common IP address - “58.84.54[.]146” - where several other previously identified domains currently resolve (Cylance Threat Research Team, 2016). The list of domains that resolved to this IP address are as follows:

Domain Name	First Seen	First Registrant Email
ssl.360antivirus[.]org	1/7/2020	Coleen.designate.56580956@gmail.com
zabbix.symanteprotection[.]com	12/4/2019	Private Registration
norton.nortonvirus[.]org	12/1/2019	Private Registration
symante.nortonvirus[.]org	10/18/2019	Private Registration
wsus.kasperskyantivirus[.]net	7/24/2017	Coleen.designate.56580956@gmail.com
update.fortinetantivirus[.]com	1/30/2017	Coleen.designate.56580956@gmail.com
dhcp.godaddydns[.]com	10/17/2016	Coleen.designate.56580956@gmail.com

Table 1: List of Domains that Resolved to 58.84.54[.]146

Introducing WLNXSPLINTER

In investigating a newly identified group that BlackBerry researchers are tracking as WLNXSPLINTER, the “moegoo[.]com” domain seen in the discussion above of the C2 infrastructure that interacted with PWNLNX3 proved interesting. It was first registered on May 3, 2009. It’s not clear whether the domain was under attacker control at this time, as it was using private registration. Over the years though, several updates were recorded with the first of investigative value occurring on February 18, 2014, using the email address void_2k@qq.com:

```
Registry Registrant ID:
Registrant Name:Wu YU
Registrant Organization:Game Develop investigation
Registrant Street:Chao Yang Road No.115
Registrant City:BeiJing
Registrant State/Province:Beijing
Registrant Postal Code:010
Registrant Country:China
Registrant Phone:67888955
Registrant Phone Ext:
Registrant Fax:67888955
Registrant Fax Ext:
Registrant Email:void_2k@qq.com
```

Figure 6: Domain Registration Information for moegoo[.]com

WLNXSPLINTER conveniently listed their organization as “Game Develop investigation” which yielded another email address that was used to register several additional domains. While BlackBerry researchers have yet to link any of these other domains to malicious activity, it is suspected that they were likely used in other previously unidentified intrusions. If you’ve seen any of the following domains in your logs, BlackBerry researchers would like to hear from you:

Domain Name	Registrant Email	Registration Date
youfunv[.]com	void_2k@qq.com	9/27/2017
heixbai[.]com	void_2k@qq.com	9/14/2017
orzk[.]net	void_2k@qq.com	4/22/2014
moegoo[.]com	void_2k@qq.com	4/28/2014
o5team[.]com	Wuyu@Tide.org	7/5/2015
moeskin[.]com	Wuyu@Tide.org	2/4/2016
is2sec[.[]com	Wuyu@Tide.org	7/2/2018
akibaol[.]com	Wuyu@Tide.org	12/20/2015
010sec[.]com	Wuyu@Tide.org	7/22/2016

Table 2: Domains Containing “Game Develop investigation” within Their Registration Information

The domain “cachecdn.moegoo[.]com” currently resolves to an IP in Google’s Cloud environment, “35.194.101.123”. The IP address was running RDP with an SSL certificate containing the issuer and common name, “chicken-01”, most likely a reference to Chinese hacking slang, Ròujī (肉鸡) (Wikipedia, 2020), where “chicken” is commonly used to denote victims of attacks.

BlackBerry researchers continue to investigate other connected activity and suspect WLNXSPLINTER’s activity was likely isolated primarily to within Asia given the similarity of many of the other domains to large Asian companies. BlackBerry researchers continue to monitor WLNXSPLINTER’s progress.

Lancer – An Installation Script

While attempting to locate subtle differences in code between PWNLNX1 and PWNLNX2, BlackBerry researchers examined the C2 protocol to assess whether any modifications were made from the earlier implants. Upon closer inspection of the custom network protocol, an additional modification was inserted into one of the implants BlackBerry researchers associate with BRONZE UNION / APT27 (based upon C2) was observed.

In this particular sample, one of the first items that will be communicated by the implant to the C2 is an encoded version of the string, “LinuxOK”. Fortunately, it turned out that this case-sensitive string was a lot less common than one would think.

After some additional digging, BlackBerry researchers identified a set of Windows PEs named “lancer.exe”. These files turned out to be the Control Panel (discussed below) used to issue commands to this particular set of backdoors. The binaries utilized the exact same XOR key - “CB2FA36AAA9541F0” – for encoding network traffic as the key observed in both families of implants.

Building on these discoveries, BlackBerry researchers identified a compressed bash shell script inside of another shell script that was responsible for installing the rootkit component of this particular variant on victim systems:

```
“e60a3a93f3930dd13b5cb115d68e4989199e366212b9809f8fc87aaa54e8e683”.
```

The initial script would write itself into a new file beginning at line 44 using the following command: “tail -n +44”. Then it would decompress the content using “gzip -cd” before executing the result. A similar script first appeared online in December of 2014 (PrudentWoo, 2014), and that original code has been reused in a number of both malicious and benign scripts.

The installer script was over 400 lines long, including comments, and referred to itself internally as “Lancer Remote Online Compilation System v2.0”, which aligned with the naming scheme used in the Windows-based controllers (lancer.exe):

```
echo "============================================================="
echo "===      Lancer Remote Online Compilation System v2.0     ==="
echo "============================================================="
echo "current system   = ${ret}bit"
echo "kernel version   = $__kernel"
echo "header version   = $__header"
echo "md5 hash version = $__version"
echo "============================================================="
```

Figure 7: The Installation Script Referred to Itself as “Lancer” Remote Online Compilation System

The script was designed to run on both CentOS/RedHat systems and Debian/Ubuntu systems. It came complete with a number of broken English phrases, misspellings, and word swaps like “being” and “begin.” The version number indicated that there was likely more than one revision of this online build system. The script additionally confirmed earlier suspicions that at least one online build environment existed for compiling and delivering the custom LKM rootkits (Group 1).

Three additional command line arguments were required for it to execute properly: a “username”, a “build”, and a third variable referenced as “force_mode”, which would force the build server to ignore an existing build for the current kernel and rebuild the LKM rootkit. The script communicated to one of two hosts depending on whether the installed kernel was 32-bit or 64-bit: “3232.3389[.]la” and “6464.3389[.]la” respectively.

A combination of command line “curl” and “wget” commands were used to interact with the remote build server. If not present, a message would be printed to the console containing the commands to install the packages via “yum” or “apt-get”. If current kernel headers were not present on the system a similar message would be printed showing how to install them. The inclusion of these messages indicated the attacker(s) using the script were likely not the creator(s). The script would first authenticate to the remote server with the supplied command line arguments similar to the pseudo request below:

```
GET /build/auth?args=username|build
```

This would be followed by a check to see if an existing LKM was already compiled for the current header version:

```
GET /build/check?args=version|kernel|force_mode&token={result from auth request}
```

If found, it would download, decompress, and install the module via an included “install.sh” script. Otherwise, the kernel headers would be compressed to /tmp/header.tar.gz and uploaded via a curl POST request to “/build/accept” with a number of additional arguments sent as form fields. Once uploaded successfully, another request would be made to the build server to compile the rootkit:

```
GET /build/compiler?args=version|kernel|version_text&token={result from auth re-
quest}
```

Once this process completed, the script would download the newly created LKM package, decompress, then install the module via the included “install.sh” script. It turned out this package of Gzip-compressed tarball would also contain the backdoors which were unique depending on which user was specified from the command line.

PWNLNX5 – The Controller

The controllers were all named “lancer.exe” and compiled on October 16, 2018. It appeared that the operator of the malware was working from a Cambodian IP address and likely testing to see if their version of the controller was detected by any industry vendors by uploading it to a common virus repository.

The threat actor similarly uploaded a tool a few days earlier that was called “OnLineTestBox.exe” that was compiled on October 12, 2018. This tool was used to simulate traffic with a C2 and allowed the user to set a couple of predefined variables to test. The tool would be useful to determine whether traffic throughput was adequate and if the receiving server was working appropriately. Here’s what the interface looked like:

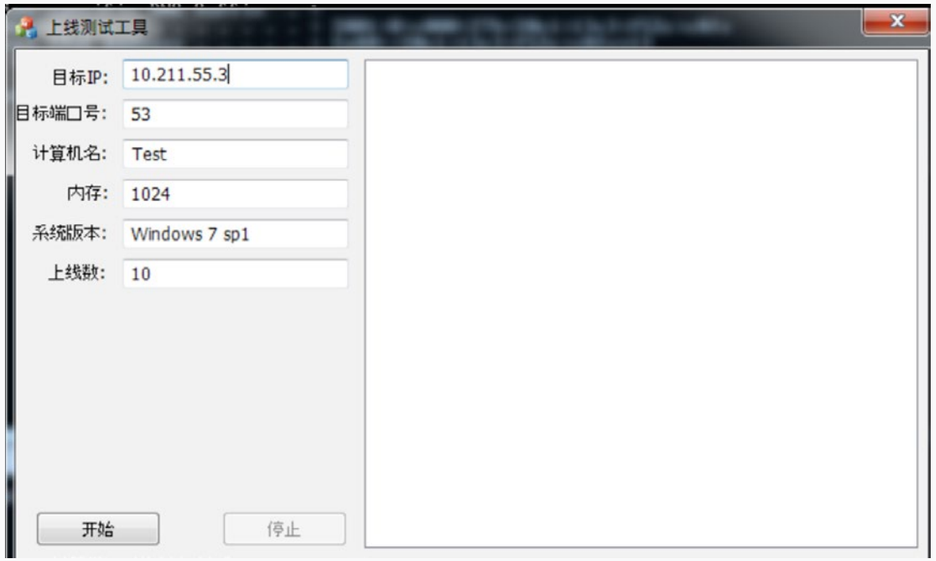

Figure 8: C2 Network Throughput Testing Utility Interface

The controller presented the operator with a nicely formatted Graphic User Interface (GUI) for controlling infected machines, and internally referred to itself as “Lancer Alpha build 2017”. The binary would look for the following files in the same directory where it was located and perform an MD5 check for each of the files listed below before starting normally:

File Name	MD5 Hash
GeoLite2-City.mmdb	7657FDB2099769206383FA59C43039F7
nc.exe	E0FB946C00B140693E3CF5DE258C22A1
puttytel.exe	146608D3DFE9F87D37EC0A41AEC2686B
Ultheme.dat	A68832233017F920B708316A007A99D9
res.zip	08708C3B17322915F286F368BB509D8C

Table 3: File Checks Performed by Lancer.exe

BlackBerry researchers were unable to fully emulate the controller, but it should be possible to patch the hash checks and reverse the format of the expected files. The code indicated that “res.zip” likely contained two additional files: “TaskMenu.xml” and “Main.xml”. After parsing these two files, several other files would be created and/or referenced, including “`\\data\info.dat`” and “`\\data\pathList.dat`”.

Several other XML configuration files were also utilized by the malware to preform various functions based upon commands issued by the operator, but they appeared to be created on-the-fly when issuing commands. Interestingly, a directory named “`\\Log\\`” would also be created within the same directory, presumably for logging purposes. The controller also contained a fair amount of content that would be outputted via debug strings.

Another Linux Oddity — CASPER Mirai Variant

BlackBerry researchers identified a single Linux Mirai variant that appeared to belong to the CASPER/LEAD group based upon the domain it communicated with: “cdn.googletoolservices[.]com”. It had the hash value:

```
57cc422a6a90c571198a2d1c3db13c31fbdb48ba2f0f4356846d6d636d0f9300
```

The researchers identified other confirmed Windows samples, each signed with a unique stolen code-signing certificate, which were communicating to another subdomain - “m.googletoolservices[.]com” — and concluded this file was connected to the group.

This backdoor curiously used the default XOR encoding method in the leaked Mirai source code developed by Paras Jha and Josiah White (Krebs, 2017). The original method takes a seed value of “0xdeadbeef” and proceeds to sequentially XOR every byte starting with the least significant and moving to the most significant. It was clear the original authors were not cryptography experts, as this method can be easily simplified into a single XOR against the byte “0x22”. The values in this compiled binary were matched to the parameters in the original source within the “table.c” file. Only the command-and-control domain had been modified within this table.

CASPER added several new features through files named “botupdate.c”, “downloadexecute.c”, “myPublic.c”, “shadowsocks.c”, and “shell.c”. Much of the functionality conveniently matched the naming conventions. “BotUpdate” provided the ability to download and execute a script via the following command: “/etc/init.d/atd start && chmod +x bot_install.sh && echo “sh bot_ install.sh” |at now +1 minutes”. The functions “DownloadFile” and “DownloadExecute” provided the ability to download and execute a file via a custom HTTP request with the following parameters:

```
GET /%s HTTP/1.1
Accept: */*
Accept-Language: zh-cn
User-Agent: Mozilla/4.0   (compatible;   MSIE   5.01;   Windows   NT   5.0)
Host: %s:%d
Connection: Close
```

The User-Agent was quite unique and contained multiple erroneous spaces. This file would be saved to “test.sh”, modified to be executable, executed, and finally deleted. A host enumeration function was also added that would write and execute a bash script file “info.sh” to enumerate the operating system and other system information.

The new “ShellExec” function provided the operators with the ability to execute shell commands. “ShadowSocksExec” and “ShadowSocksClose” provided the ability to install and uninstall ShadowSocks, a secure SOCKS5 proxy, via a script “ss.sh” (SDT, 2019). While it was unclear where this script would be derived from, it was most likely similar to the one available at the following URL: “http://blog.whsir.com/uploads/ss.sh”. The script would install ShadowSocks and all required dependencies through a single click on multiple operating systems.

Having now discovered and mapped out a previously undisclosed Linux toolset, as well as the support environment used to build and run it, BlackBerry researchers turned their focused to another often-neglected attack vector: mobile devices. Their first find was related to the Android operating system, which shouldn’t come as too much of a surprise since 80% of the Android kernel is based on Linux. Yet what they ended up discovering nevertheless shattered all expectations.

Cellular Division

In a previous research cited in the beginning of report, *Mobile Malware and APT Espionage*, BlackBerry researchers provided evidence that RATs designed for mobile devices, primarily those on the Android platform, have been developed and deployed by APT groups working in the interest of the Chinese government for far longer than had been publicly acknowledged. Included among those groups was at least one that had been previously associated with the WINNTI GROUP.

Upon closer examination of the groups leveraging the Linux implants, BlackBerry researchers found a number of indications within current and older C2 infrastructures that mobile implants associated with both PASSCV and CASPER likely existed.

PWNDROID4

An example of one such domain was “ios.0pengl[.]com”, which resolved to the IP address “122.226.186[.]28” beginning on November 25, 2015. BlackBerry researchers identified several other subdomains that were potentially of interest and went looking for the associated malware.

Their findings didn’t bear out exactly as anticipated, however they did discover a previously unattributed Android malware sample capable of monitoring incoming/outgoing phone calls, recording audio, sending and receiving SMS messages, and monitoring a device’s GPS location. BlackBerry researchers designated this new Android backdoor PWNDROID4.

The package the researchers identified was named “com.wavedancer.host” with the SHA256 hash:

```
ac546bd38ad2e56b42fd3e35f27048ca9c86203153868944188e6fb6822d9f63
```

It was likely created on June 16, 2015, based upon the last modification time of the APK package contents. Very little effort was taken to obscure information or code within the application, and it appeared to be a first-generation test by the attackers given the internal name “GPS_TEST”.

The Android Manifest file is a great place to start when taking a look at any APK file; it is located within the root directory of the APK and will have the filename “AndroidManifest.xml”. The manifest granted the application a broad range of permissions:

android.permission.ACCESS_FINE_LOCATION	android.permission.READ_PHONE_STATE
android.permission.ACCESS_NETWORK_STATE	android.permission.READ_SMS
android.permission.ACCESS_WIFI_STATE	android.permission.RECEIVE_BOOT_COMPLETED
android.permission.INTERNET	android.permission.RECORD_AUDIO
android.permission.MODIFY_AUDIO_SETTINGS	android.permission.SEND_SMS
android.permission.WAKE_LOCK	android.permission.READ_CALL_LOG
android.permission.READ_CONTACTS	
android.permission.PROCESS_OUTGOING_CALLS	
android.permission.WRITE_EXTERNAL_STORAGE	

While this is a frightening list of permissions, this particular malicious APK would most likely need to be sideloaded and/or downloaded by a slightly less intimidating application. The “Receive Boot Completed” permission is the equivalent of letting the application know when the device has been restarted, which almost always guarantees it contains some form of persistence upon reboot.

The manifest additionally contains a “minSdkVersion” and “targetSdkVersion” which gives some additional time-based context. They provide which versions of Android the application was designed to run on (Android, 2020):

```
<uses-sdk
    android:minSdkVersion=”10”
    android:targetSdkVersion=”22” />
```

Figure 9: SDK Version Information from AndroidManifest.xml

This marker indicated that the application was designed to run on Android versions Gingerbread (2.3.3) to Lollipop (5.1). Android Lollipop 5.1 was first released in March of 2015, which nicely aligned with the last modification date identified from the contents of the APK. The standard resources file “resources.arsc” was readily parsed and yielded immediately useful information:

```
<?xml version=”1.0” encoding=”utf-8”?>
<resources>
    <string name=”app_name”>GPS_TEST</string>
    <string name=”connections_list”>ns6.0pendns.org:53;ns6.0pendns.
org:80;ns6.0pendns.org:443;</string>
    <string name=”hide_launcher_icon”>0001</string>
    <string name=”host_id” formatted=”false”>Host-%Rand%</string>
    <string name=”auth_password”>Password</string>
    <string name=”connection_delay”>15</string>
    <string name=”connection_type”>00000001</string>
    <string name=”proxy_list”>0:10.20.30.40:1000;0:200.215.14.62:1080;0:100.200.
14.62:1080;0:10.20.30.40:1000;0:200.215.14.62:1080;0:100.200.14.62:1080;</string>
</resources>
```

Figure 10: XML Strings Parsed from “resources.arsc”

The backdoor would attempt to connect to “ns6.0pendns[.]org” on ports 53, 80, and 443. It may also potentially utilize one or more proxies: “10.20.30[.]40:1000”, “200.215.14[.]62:1080”, and “100.200.14[.]62:1080”. Both of the latter IP’s immediately show up in open SOCKS v5 proxy lists, which probably makes them of little investigative use. The first historic resolution BlackBerry researchers could find for the C2 domain “ns6.0pendns[.]org” was in March of 2019: “150.242.210[.]158”.

This IP was probably not all that relevant, as the implant was likely only active four years prior. The domain “0pendns[.]org” was first registered using the email address “timew4lk@gmail.com” which connected to other PASSCV infrastructure as previously mentioned above.

An Interesting Find

After digging into the decompiled code of PWNDROID4, BlackBerry researchers found striking structural layout similarities and identical swathes of code that appeared to match the Android version of NetWire - a find worth digging into:

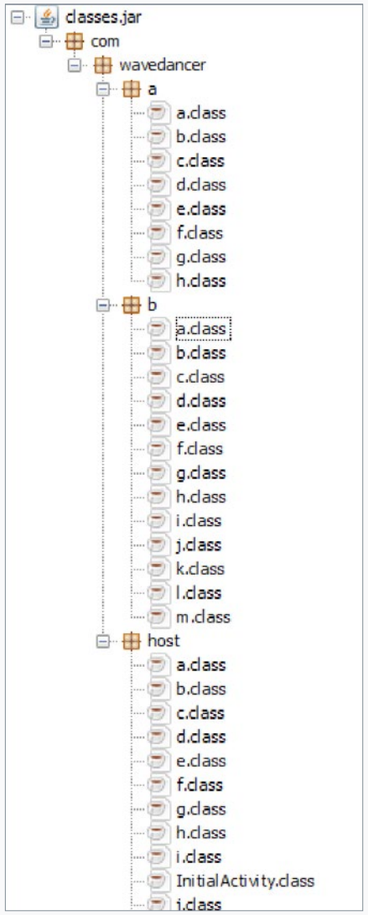

Figure 11: PassCV Code Layout

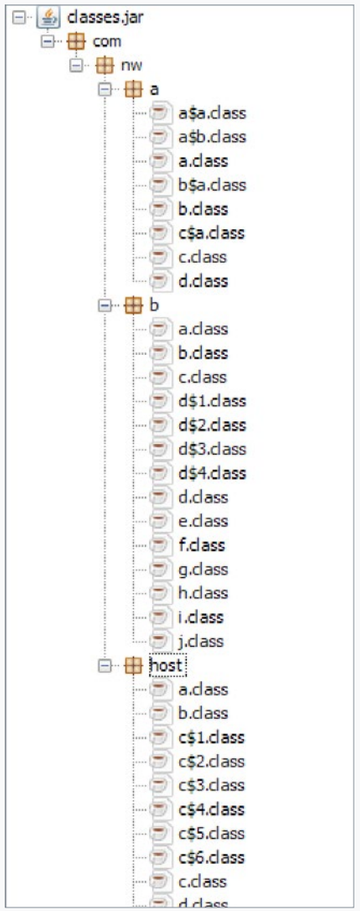

Figure 12: NetWire Code Layout

NetWire is a multi-platform, commercial, off-the-shelf remote administration tool (RAT) that can be licensed on a monthly or annual basis from a company called World Wired Labs (https://www.worldwiredlabs.com). It is marketed for legal use by systems administrators, incident responders and - curiously enough – parents who want to monitor their kids' mobile phone activity. World Wired Labs says that "NetWire can be customized to suit your daily needs, such as remote support, live forensics or even monitoring your children at home" (World Wired Labs, 2020).

But security researchers will immediately recognize NetWire as one of the most pervasive RATs in use by criminal enterprises and APT groups, but that's not the fault of World Wired Labs: tools are tools and it is the intent of the user, the authorization of the target, and the laws of the applicable jurisdictions that are the discriminating factors when determining whether a tool is being used for good or for nefarious purposes.

The abuse of publicly available hacking tools by adversaries has been pointed out by the governments of the so-called Five Eyes nations (U.S., U.K., Canada, Australia, New Zealand) in a joint report released in 2018. The report cautioned: "Experience from all of our countries makes it clear that, while cyber actors continue to develop their capabilities, they still make use of established tools and techniques. Even the most sophisticated groups use publicly available tools to achieve their objectives" (National Cyber Security Centre, 2018).

A close look at the World Wired Labs website does not yield the kind of contact information one might expect to see for a company interested in selling its offerings in the open market. While the site offers three ways to contact the company – via live chat, emailing support@ or filling in an online form – there is only one person identifiable at the company which appears at the end of each press release, but with no contact information. There is no phone number listed for the business, and no address is provided aside from a Google Map snippet showing an imprecise office location somewhere in Belize.

At first, BlackBerry researchers speculated that PASSCV was simply trying to adopt an existing backdoor into their repertoire - as noted above, it is not uncommon to see APT groups adopt publicly available hacking tools. Here's where it got a little strange though: the first public announcement by World Wired Labs about forthcoming support for the Android operating system was made on January 2, 2017, nearly eighteen months after PASSCV's PWNDROID4 was created based on an archived screenshot of the "Android Support" announcement on the NetWire Website (The Internet Archive, 2020).

A second announcement was made on March 13, 2017, indicating that the Android release would be delayed. Finally, NetWire Version v1.7a was released on March 23, 2017, marking the first public release of the Android host - nearly two years after the eerily similar PASSCV malware was created. The remarkable overlap in structure and coding combined with the timeline for the development of the tools certainly raises some questions about their connection. BlackBerry researchers were unable to locate any similar code samples across their archives or anywhere in public and semi-public domains.

BlackBerry researchers also could not locate an iOS® implant used by PASSCV, however they strongly suspect that one or more may be out there, as a lot of the other early subdomains this group used were overly descriptive and have turned out to be reflective of reality. Given the unsophisticated nature of the Android implant and the timeframe of 2015, BlackBerry researchers suspect if any iOS implants exist, they likely would have only executed on jailbroken devices or been delivered alongside of jailbreaking tools.

CASPER Goes Mobile – PWNDROID5

BlackBerry researchers identified several implants designated as PWNDROID5 which masqueraded as fake Adobe Flash updates for Android in a newly identified campaign designated as OPERATION ANDROIDBEACON. Most were deployed in early to mid-2016. Each made network requests to subdomains under associated CASPER/LEAD C2 infrastructure.

On first glance the APKs appeared completely benign. But malicious APKs were encrypted and stored within the "assets" folder using a random six-character string. These inner APKs were encrypted using a standard DES cipher with an arbitrary key. For example, in the case of this sample:

```
64424a7c5f0d8e1c5d64c4c6fa9bdc2987dbdcf1bafdb6f45df9e783712c5187
```

The DES key was "F4o6VdRP". The actual code to decrypt the APK was not written inside this particular DEX file and was instead compiled into a standard Linux library named "libentry.so". The library was compiled for ARM as well as x86 and stored within the respectively named folders inside the standard "lib" folder. Below is a python function to decrypt these inner APKs. The key will always be 8-bytes in length and may need to be modified based upon the contents of the external library, in this case "libentry.so":

```
from Crypto.Cipher import DES
def decrypt_apk(buf):
     key = "F4o6VdRP"
     des = DES.new(key)
     return des.decrypt(buf)
```

Figure 13: Python Function to Decode CASPER APK

Many of the APKs identified used some variation of this, and either had the decryption code embedded within the actual APK or in an external library as in the above sample. The decrypted APKs would make a POST request to one of the following base URLs:

```
http://app.appleadwords[.]net/s1/
http://app.appleadwords[.]net/s2/
```

A number of additional parameters would be sent along with the base URL:

Variable Name	Value
pn	Package Name
vc	Version Code of the Package
md	Build Model
ov	Version Release
mc	Mac Address
lc	Locale
chn	Set to ofw
did	Device ID
anid	Android ID
refer	Set to Empty by Default
sys	Either 1 or 0 depending if application is installed in device's system image

Table 4: Parameters Sent in CASPER Network Check-In

BlackBerry researchers were unable to retrieve the follow-on payload(s) given the amount of time that had elapsed before discovery. It appeared that these payloads would likely be AES encrypted DEX or JAR files with the key "hello@#fe931AaBb" and use an initialization vector "0102030405060708".

The payloads could also potentially be wrapped in an extra layer of DES encryption with the key "password". BlackBerry did identify a writeup by Sophos on a sample that appeared to be related, one they termed "Andr/Axent-DS" (SophosLabs, 2017). Upon looking into the IP address "114.108.185[.]113", it was clear these were all likely part of the same family. With the ones beaconing to "app.appleadwords[.]net" starting about six months after the domain Sophos identified: "s1.deepcups[.]com". The domains "psserviceonline[.]com" and associated subdomains as well as "app.aqmobi[.]com" and "bht.aqmobi[.]com" also appeared to be related to the same family of Android malware and used a similar request structure.

BlackBerry researchers determined this family likely survived and now makes its home on

Amazon AWS infrastructure using a variety of different domains:

```
asense[.]in
mobnativeads[.]com
mobileflyx[.]com
mydataprovider[.]in
napiservice[.]com
native123[.]com
nativeload[.]com
natureapi[.]com
nsdknative[.]com
ntracecloud[.]com
p2nservice[.]com
pdbarea[.]com
sdatareport[.]com
subclicktrack[.]com
tnapiservice[.]com
```

If a mobile device is in fact beaconing to one of these domains associated with OPERATION ANDROIDBEACON, you may want to take a closer look.

Windows Base Camp

Windows has historically provided a free seaside beachfront vacation home for APT groups associated with the WINNTI approach. That's because their Windows RATs have targeted both desktops and servers seamlessly in their operations for more than a decade. But, as we have seen, some of these groups have also demonstrated expertise in attacking Linux and other platforms. So how, then, might the same groups go about managing an attack against both platforms simultaneously? As it turns out, it's not that hard.

While pivoting between Linux and Windows platforms may seem like a novel phenomenon, several of these groups were documented doing just that way back in 2012 (Fraser, et al., 2019). More recently, one of the groups that that CrowdStrike tracks as WICKED SPIDER has been observed targeting MacOS as well (Bradley, 2018).

In order to make moving between platforms easier, several of these APT groups have employed the cross-platform tunneling tool EarthWorm to proxy traffic between different operating systems (idlefire, 2016). EarthWorm was first released by "rootkiter" by a security researcher at China's Qihoo 360 Netlab in May of 2015 (Rootkiter, 2015).

Many of the previously exposed WINNTI Windows implants have continued to evolve, and the APT groups behind them continue to deploy new malware alongside other well-known families as well as other signed, open-source backdoors.

One common feature going back through the years shared by all the groups collectively referred to as WINNTI has been the use of stolen code-signing certificates. Typically, these code-signing certificates belonged to compromised video game developers. The stolen certificates were subsequently used to sign malware deployed in attacks on other higher priority targets, such as SK Telecom in 2011 (Command Five Pty Ltd, 2011).

However, BlackBerry researchers discovered what appears to be a novel and ongoing trend: the threat actors have shifted from compromising video game companies to compromising adware developers, and then utilizing their code-signing certificates in operations.

Adware? Who Cares?

At first glance, using code-signing certificates belonging to adware developers seems completely counterproductive. Malware that may previously have gone undetected would now almost surely be immediately noticed. At least a handful of antivirus vendors would flag it, if only on the basis of the adware code-signing certificate. Why would an attacker, particularly one aligned with the interests of a nation state, want to do that? See Figure 14:

Many of the previously exposed WINNTI Windows implants have continued to evolve, and the APT groups behind them continue to deploy new malware alongside other well-known families as well as other signed, open-source backdoors.

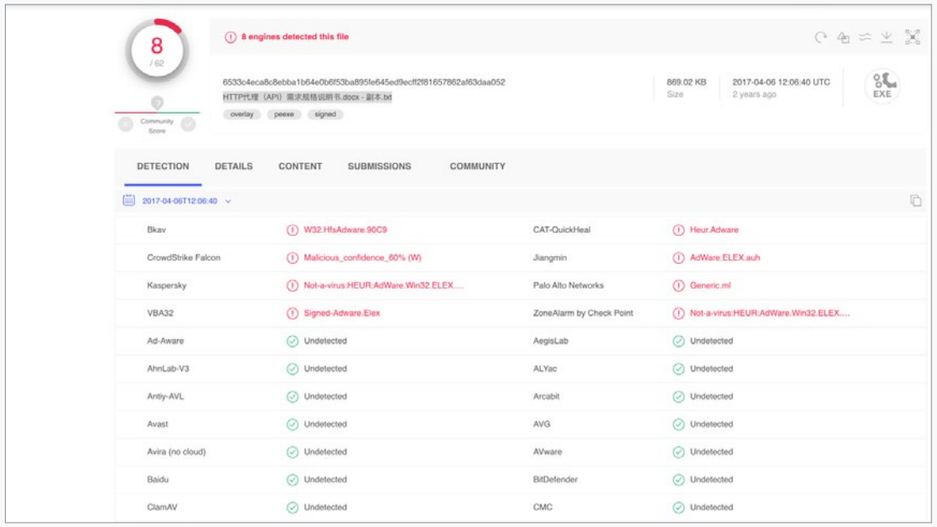

Figure 14: Example of Malware Signed with Adware Certificate and Detections

In our judgement, these threat actors would rather be found and then ignored than found and investigated, particularly on the Windows platform where so much of the antivirus attention is focused. Malware masquerading as adware stands a good chance of being overlooked or disregarded if it is detected, especially in busy corporate enterprise environments because they manage a “stack” of multiple security technologies, each with its own set of alerts.

While this practice often provides greater visibility, it can also obscure what really matters. The APT groups BlackBerry researchers investigated in this report seem to have leveraged this. Many of CASPER’s samples are currently and have historically been flagged by numerous antivirus solutions as potentially unwanted applications (PUAs), which is an understatement to say the least.

Analysts, if presented with a piece of information, tend to use it. So, if an antivirus program alerts an analyst to adware on a system, they would tend to trust that classification. How likely would you be to investigate further once that happened? BlackBerry researchers suspected the action for most would be to simply move on. After all, a large majority of adware is signed using legitimate code-signing certificates anyway.

Both network and host defenders are usually inundated with alerts on any given day, and filtering through them can be a monumental task unto itself. Determining which of those adware alerts is actually the foothold for an APT intrusion and not just a run-of-the-mill nuisance is an unreasonable expectation for the average organization.

BlackBerry researchers believe these types of modifications will become increasingly common as time goes on. It’s likely that malware bearing stolen adware certificates will need a new classification or designation within the information security community so it can be better understood.

What the attackers have done in donning the façade of adware is to directly target the psychology and methodology of blue team members to exploit inherent weaknesses in their assumptions. Alert fatigue is real, and adware is boring. The authors of this paper have increasingly seen these techniques employed by a number of other nation state actors to intentionally avoid analysis, or at the very least provide a layer of misdirection that’s not easily detected. Readers will find a list of compromised adware and greyware code-signing certificates and associated malicious binaries in the Appendix of this report.

New(er) Droppers:
In this section, we undertake a technical analysis of several of the newer, and thus, undocumented updates to the modified ZXShell variants commonly used by BRONZE UNION (aka APT27, EMISSARY PANDA).

General readers will note that all of these new droppers were predominantly signed with legitimate certificates stolen from other companies based in Asia and adware vendors. This may indicate either a tactical shift in targeting priorities or may simply be a case of harvesting the lowest-hanging fruit.

Examination of several newer ZXShell droppers, ones which BlackBerry researchers associate with BRONZE UNION, began by noting changes made following Dell's report in February of 2019 (Dell Counter Threat Unit Research Team, 2019). After public research is released, threat actors often change their tactics. BlackBerry researchers took a closer look to see what, if anything, had changed in response to the publication of that research.

The new droppers examined were signed with older stolen certificates, however the rootkits used a previously unseen code-signing certificate belonging to "Xiamen Tongbu Networks". Interestingly, several of the other groups discussed in this report also continued to use other older stolen certificates and countersigned their executables so they would remain valid long past the certificates' original expiration date. Some examples:

Recent Droppers:

- ce3424524fd1f482a0339a3f92e440532cff97c104769837fa6ae52869013558
- caa46c001c3180eb7fdd5e5cbf7d084b75b7bdf72e61e06430a88378604a25eb
- fbe294910ef833e1c9b2c8663c06b6ef99c13b2bc5eb01e87defb798c8066f0b
- 8674c76583c13c60fcb6dc344bae4a5149cce35a85bb600f0a6af5e769b98585

BlackBerry researchers examined whether anything substantial had changed from 2018 to 2019, starting with the SHA256 hash beginning "ce34" above (#1 in the list).

Dropper 1
The sample was a 32-bit executable with a compile date of "January 1, 1970 03:25:45am UTC" – an indication to that a packer may have been used to obfuscate the malware and impede detection. However, this was a somewhat unusual time and was not associated with any particular programming languages or known packers as far as the researchers were aware. The actual UNIX timestamp value equated to "12345".

At first glance it appeared that the file was packed with UPX, based upon the section names ".UPX0" and ".UPX1". This was clearly untrue though, given the sheer number of additional anti-debugging tricks it employed. In any case, these seemingly insignificant details when taken together were used to identify related files despite not getting to the bottom of the packer mystery.

The following YARA rule will find all related malware packed in this manner:

```
import "pe"
rule BronzeUnionPacker
{
    condition:
        pe.timestamp == 12345 and for any i in (0..pe.number_of_sections - 1):
(pe.sections[i].name == ".UPX0") and pe.number_of_signatures >= 1
}
```

Figure 15: Generic Yara Detection for Bronze Union Packer

Once unpacked, the dropper would create two files: a driver with the full path "\\Windows\\System32\drivers\ autochk.sys" and a DLL in one of the following locations:

\\Windows\\System32\\AudioSdk.dll	\\Windows\\System32\\cryptdns.dll	\\Windows\\System32\\odbcwg32.cpl
\\Windows\\System32\\audiosrc.dll	\\Windows\\System32\\dhcpcsvcd.dll	\\Windows\\System32\\PINTLGNT.dll
\\Windows\\System32\\bitsprx.ime	\\Windows\\System32\\imekr61.dll	\\Windows\\System32\\prnfsdk.dll
\\Windows\\System32\\bootred.dll	\\Windows\\System32\\imseo21.ime	\\Windows\\System32\\samlib32.dll
\\Windows\\System32\\C_1950.NLS	\\Windows\\System32\\iscsiapi.dll	\\Windows\\System32\\shlwapi.dll
\\Windows\\System32\\c_21268.nls	\\Windows\\System32\\KBDDWSKY.DLL	\\Windows\\System32\\shlyapi.dll
\\Windows\\System32\\C_26849.NLS	\\Windows\\System32\\keyzip.dll	\\Windows\\System32\\shlzapi.dll
\\Windows\\System32\\chrsben.dll	\\Windows\\System32\\mfc100usx.dll	\\Windows\\System32\\sqlnclc11.dll
\\Windows\\System32\\chrsben.ime	\\Windows\\System32\\mfc120du.dll	\\Windows\\System32\\stdole32.dll
\\Windows\\System32\\cliconfg.cpl	\\Windows\\System32\\midiapi.dll	\\Windows\\System32\\wbem\\loadperf.dll
\\Windows\\System32\\cryptbios.dll	\\Windows\\System32\\odbccx32.dll	\\Windows\\System32\\wlanseo.dll

Table 5: Potential FilePaths for the ZXShell Backdoor

```
Driver:     28924b6329f5410a5cca30f3530a3fb8a97c23c9509a192f2092cbdf139a91d8
DLL: a37574387a4bacfb69e7369d6ac8749603038a1b232d9a482bbcd2dce0c091b0
```

The driver prevented deletion of the backdoor while it was loaded and redirected any file-based requests made of the backdoor to the legitimate "shlwapi.dll". It would similarly redirect file requests made of the driver "autochk.sys" to the legitimate "fltMgr.sys". When the researchers arrived at this discovery, they realized that Ori Damari had already produced an excellent writeup containing this same revelation. His detailed report was produced in November of 2019 (Damari, 2019).

That said, the rootkit for the particular sample BlackBerry researchers examined lacked any of the network-hiding features Damari wrote about. Instead, it curiously contained two different code-signing signatures: one belonging to "Shanghai Hintsoft Co., Ltd." using a SHA1 digest, and the other belonging to "Hangzhou Bianfeng Networking technology Co., Ltd." using a SHA256 digest. Details of both certificates are listed in the appendix of this report. The DLL was similarly signed with the "Hangzhou Bianfeng" certificate "June 6, 2018 4:49:06 UTC", which provided a better approximate idea of when the dropper was actually created because the compile time of the DLL was "April 10, 2018 19:42:09 UTC".

The DLL exported a number of unique functions that could be readily used to identify similar backdoors. The dropper would configure the DLL to run as a ServiceDLL on boot beneath a new randomly named service beginning with the string "netsvc_" and ending with eight random lowercase hex characters. Analysis of the DLL was hindered slightly by the custom packer but dumping from memory worked like a charm. The DLL was a modified variant of ZXShell and quite large. The backdoor contained a plethora of functionality and could accept the following commands:

Help	Displayed Help Information
Exit/Quit	
Sysinfo	Provided System Information including OS, disk information, CPU, RAM, user information, and system uptime
RunAs	Executed a process as another user
GetCMD	Provided an interactive Command Shell
SockProxy	Started a Socks4 or Socks5 Proxy via internal program “SockProxy V1.2”
PortScan	Port Scanning Functionality
ShareShell	Shared a shell to others via netcat-like function
SuspendFW	Suspend the Windows Firewall
Ps	Provided process and service management
FileTime	Cloned a file’s timestamp information
fileMG	Provide an interactive file manager
winvnc	Remote Desktop (did not appear to be used)
rPortMap	Remap a port on the local host
Remarks	A commenting system
logonPasswords	Dump cleartext stored passwords
Htran	Full Htran functionality
Uninstall	Uninstall and remove the DLL

The group used a modified cipher for sensitive parameters, such as network callback information that utilized a combination of base64 and a custom XOR implementation. The cipher stored the size of the string in the last value and used it as a seed value to generate a custom position dependent XOR key.

Dropper 1 Network Callback Details

The backdoor would attempt to beacon to the domain “tdjsyqty0takah2x.gitoos[.]com” on ports 53, 80, and 443. The domain currently resolves to the IP address “35.186.159[.]221” which belongs to Google’s cloud service “Google Compute Engine™.” BlackBerry researchers identified several other domains ascribed to BRONZE UNION that currently resolved to IP addresses within Google’s cloud service. All of the IP addresses were running the Remote Desktop Protocol (RDP) Service on TCP port 3389 and appeared to be various Windows virtual machines:

```
Initial Handshake:
00000000 04 c8 07 d8 30 b9 03 98 da ac 7b 10 13 20 03 10    ....0... ..{.. ..
00000010 fc 0b 53 0f                                        ..S.
00000000 12 b0 11 30 83 98 19 10 57 55 1d 50 82 99 39 10     ...0.... WU.P..9.
00000010 7f 35 ec 79                                        .5.y
Info Request:
00000014 fd 81 4e 00 98 5e a2 60 ac 64 b2 60 bc 1c ba 10     ..N..^.` .d.`....
00000024 fc 18 45 05                                         ..E.
00000014 dc 8f 6b 1c 98 8b 27 18 54 87 e3 14 10 83 bf 10 .  .k...’.
T.......
00000024 7e 42 de 6f                                          ~B.o
00000028 55 73 65 72 2d 50 43 40 2d 61 64 6d 69 6e 40 31    User-PC@ -admin@1
00000038 39 32 2e 31 36 38 2e 31 30 30 2e 33 30 20 4f 53    92.168.1 00.30 OS
00000048 3a 20 57 69 6e 64 6f 77 73 37 20 50 72 6f 20 53    :Window s7 Pro S
00000058 50 31 2e 30 28 37 36 30 31 29 20 43 50 55 3a 33    P1.0(760 1) CPU:3
00000068 36 38 34 20 4d 48 7a 2c 33 36 31 36 20 4d 48 7a    684 MHz, 3616 MHz
00000078 2c 33 36 31 30 20 4d 48 7a 2c 33 36 30 30 20 4d    3610 MH z,3600 M
00000088 48 7a 2c 52 41 4d 3a 34 30 39 36 4d 42             Hz,RAM:4 096MB
00000028 f4
00000095 0d 0a 55 73 65 72 2d 50 43 3e 0d 0a                ..User-P C>..
```

Figure 16: Network Traffic exchange from Modified ZXShell Protocol

The hex dump above shows the initial network traffic exchange between the victim and server. The server's traffic is in blue while the victim's traffic is in red. The protocol was modified from the original source, so the researchers did not spend too much additional time on it. Identification of servers on the internet that send data first on port 80, 443, or 53 should be a dead giveaway that something is not quite right. The initial exchange always used TCP packets that were 0x14 bytes in length as opposed to 0x10 bytes as in the original protocol.

Dropper 2:
The hash beginning "caa" (#2 in the list above) was a much newer dropper signed on July 26, 2019, using the "Hangzhou Bianfeng" certificate previously mentioned, and it was packed with the custom packer described above. It utilized the exact same file locations for the driver and DLL, although both dropped files bore different hashes:

```
Driver: 9b7c1e37d5f56cc0b5e5e22ce9805e237a189297e78405b9c392a0953b6e0321
DLL: 101171cc6ffda3428089e77ce2a90f0d2f490fa68970c09f777c5ec0b0707cf6
```

BlackBerry researchers took a closer look to see what if anything had changed from 2018 and found that driver was signed with a new stolen code-signing certificate belonging to "Xiamen Tongbu Networks Ltd.", which was founded by the former head of Google China, Li Kai Fu.

BlackBerry researchers identified another rootkit signed with the same driver, which is also listed in the Appendix. The file's compile time appeared to be accurate, as it was "May 4, 2019 21:34:08 UTC", just before the DLL was compiled. The driver's code had been updated and the list of filenames was now stored with the characters reversed, but otherwise the paths remained the same. It contained new code which provided the ability to hide network connections made by the backdoor from tools like netstat, as explained in the Damari research (Damari, 2019).

The first noticeable difference between the two DLLs was that the compile date of the second DLL seemed to be accurate: "May 4, 2019 21:36:36 UTC". The file contained UPX section names ".UPX0" and ".UPX1" and was still protected with the same custom packer. The researchers dumped and rebuilt the DLL from memory to speed up analysis. The DLL was more or less unchanged from a command and functionality perspective, and it used the same custom cipher to obfuscate sensitive strings and a similar modified network protocol with 0x14 byte handshake.

Dropper 2 Network Callback Information
The ZXShell variant dropped by the second sample beaconed to a different domain: "yofeopxuuehixwmj.redhatupdater[.]com" on TCP ports 53, 80, and 443. The first recorded IP resolution the researchers could find was in September of 2019, nearly four months after the backdoor was built.

The IP addresses the domain resolved to did not appear to coincide with any similar samples. They belonged to VeeSP (https://www.veesp[.]com/en), Fishnet Communications, and Profit Server (https://profitserver[.]ru/en). This may represent a tactical shift for newer domains or indicated a disparate or separate attack group may have been responsible. One IP range was of particular interest, "77.73.64[.]0/21", as other attack groups including CHAFER (Symantec Threat Intelligence, 2018) had utilized IP addresses within it.

As the intellectual property and other targeted data has moved to new operating environments, these groups have readily adapted, shared new tools, borrowed from open-source resources, and developed new methods to harvest information - all while effectively hiding more or less in plain sight.

The Bigger Picture - Network Infrastructure

The use of cloud environments represented a change in TTP's for the PASSCV, CASPER, and BRONZE UNION groups. Cloud servers provide an ideal C2 environment for malware operators because they can easily be moved, easily be deployed, and easily be managed in contrast to more traditional virtual private servers (VPS's) or dedicated servers.

BlackBerry researchers discovered that the majority of servers used by these groups for C2 currently reside within Google's cloud service. Some details about the current as well as historical activity by both BRONZE UNION and PASSCV are provided below:

Google Cloud™ Linux C2 IP Addresses:

```
35.185.156[.]217
104.199.158[.]58 - Rootkit Build Server
35.185.188[.]253
35.186.158[.]135
35.186.159[.]111
35.194.101[.]123
35.201.147[.]249
35.234.57[.]84
35.236.143[.]199
35.236.181[.]31
```

Google Cloud Windows C2 IP Addresses:

```
35.185.185[.]214
104.199.173[.]2
34.80.77[.]57
35.186.159[.]221
35.187.155[.]1
35.187.194[.]33
35.194.170[.]0
35.187.215[.]226
35.187.217[.]64
```

Unknown Google Cloud C2 IP Addresses

```
35.185.189[.]30
104.199.235[.]60
```

Curiously, it appeared that one of the operators may have made an error, or conversely, used the same virtual machine on two different IP addresses. BlackBerry researchers were able to locate an SSL certificate with a common name of "windows-15" and a serial number of "350591961586887474315324461082510744 37" that was used on both "35.187.155[.]222" (Google Cloud) and "58.84.54[.]147", an IP address in Hong Kong. Several other C2 servers were also located within the 58.84.54[.]0/24 net block. The groups additionally deployed Tencent Cloud and Alibaba Cloud servers to a lesser extent.

Attribution

WINNTI began as a backdoor (Symantec, 2011), then it was designated as a group (Kaspersky Lab Global Research and Analysis Team, 2013), and later it was identified as an “Umbrella” (Hegel, 2018). Today, it’s become something of a threat intelligence analyst’s nightmare.

We should have expected this to happen because every security group has different data sets and analysts of varying abilities at their disposal, all of which results in a vastly different view of the proverbial “elephant in the room.” But it hasn’t stopped these analysts from trying to make connections to others’ research, however tenuous. As a result, in some readings of APT security research focused on groups acting in the interest of China, WINNTI seems to be everywhere with seemingly every group investigated somehow connected to it. This is unhelpful.

It’s worth noting that while early WINNTI-related malware and infrastructure may still be around, the people behind it have almost certainly come and gone. To think that the original WINNTI GROUP, as defined by Kaspersky in 2013, is somehow still together all these years later is wishful thinking. So, when readers encounter WINNTI in the press or in research, how should they understand it? What does the designation signify today?

The WINNTI Approach

The researchers’ considered opinion here is that WINNTI has come to represent more of an approach rather than a moniker for any single crew. It refers to a method of attacks wherein cells of civilian contractors are assembled, attack tools and intelligence are shared, and the targets are assigned.

The tools and infrastructure favored by each cell, or APT group, differs - but sharing between the groups regularly occurs. This suggests that either the APT contractor community in China regards sharing favorably and tolerates it openly, or that members travel between cells over time or groups of them break off from individual cells to form new ones, or both. It is also possible that the Chinese government, which is assessed to be their likely customer, provides something in the way of support by providing tools and intelligence in some formal fashion, but this is the least likely scenario.

The use of stolen code-signing certificates - typically from video game companies, but as we have discussed in this report, now too from adware companies - is another common bond. Kaspersky researchers did well to point out the common criminal ancestry of the original WINNTI GROUP. This criminal legacy to operations has continued to color those of the original group’s descendants all these years later, particularly in their wide and voluminous targeting. It looks like “spray and pray,” but it’s more likely done with more strategic intent.

The majority of the groups discussed in this report - PASSCV, WINNTI GROUP, CASPER (LEAD), and BRONZE UNION - have been discussed in other public security research. Whether these groups were actively collaborating, casually sharing, comprised of some of the same members, or in actuality were smaller parts of some larger group is beyond analysis at this point. Occasionally, though, cracks appear that provide some greater insights.

The longevity of the WINNTI approach and its non-government attacker-culture ancestry has meant that mistakes in these groups’ operational security have frequently come to light. Researchers at TrendMicro, BlueCoat (now Symantec, whom we credit with finding and naming of PASSCV), Dell SecureWorks, ESET, and Kaspersky have all spilled ink about chasing the odd bits of personal information that have fallen through those cracks. Let’s look at one of the more interesting ones:

In 2013, Kaspersky researchers identified one of the original suspected WINNTI GROUP members by his screenname “Mer4en7y” as the result of an apparent lapse in operational security (Kaspersky Lab Global Research and Analysis Team, 2013).

In tracing the attacker's online footprint circa 2012, they found that they were a member of a hacking forum, had submitted a vulnerability in a commercial bank system, maintained a microblogging page, and was part of an information security forum called "90 Security Team." As Kaspersky pointed out, they were based in Nanjing and once posted a reply to an ad looking for "powerful pentesters" in another city, writing "aren't you recruiting people for APT? Guangzhou is too far, but anyway I support it."

As fate would have it, Mer4en7y's moniker appeared again in print in another writeup on WINNTI malware. This time, though, it was a U.S. Department of Justice indictment. Prosecutors charged Mer4en7y for hacking Capstone Turbine in what they said was an MSS conspiracy to engage in computer network exploitation operations in furtherance of corporate espionage (United States of America v. Zhang et al, 2017).

Researchers don't typically have the resources of a government, nor the inclination to assess whether the Mer4en7y connection is what it seems, but it's worth pointing out if only to highlight what prosecutors alleged was the way in which the attackers organized. They said an intelligence official from the provincial branch of the Ministry of State Security (MSS) had recruited a bunch of civilian hackers like Mer4en7y – mercenaries, if you will – to carry out the mission.

If you believe what the authors behind a number of "Intrusion Truth" (https://intrusiontruth.wordpress.com/) posts have written about APT groups acting in the interests of the Chinese government, this all fits a very distinct pattern, one where missions and directives change over time yet have a common theme with a regional intelligence officer directing a local network of contractors to achieve a longer-term strategic goal with significant flexibility in who is employed and how the objectives are carried out.

Conclusion

It should come as no surprise that a number of the groups affiliated with the WINNTI approach have continued to effectively compromise their targets over the past decade. The groups have rapidly adapted to changes in defenders' tactics and continually evolved their toolsets and techniques as the target landscape has changed.

As the intellectual property and other targeted data has moved to new operating environments, these groups have readily adapted, shared new tools, borrowed from open-source resources, and developed new methods to harvest information - all while effectively hiding more or less in plain sight. In addition, many of the attack techniques that worked a decade ago continue to be effective today. The cycle regularly comes full circle, where old techniques and tricks are revived time and time again.

While much of the security industry continues to charge forward with efforts to address the next trendy buzzword threat, few are looking back in time to assure they have effectively solved for the issues presented by the last. Thus, some subtle changes in tactic and a new stolen code-signing certificate appear to be the only things necessary for these adversaries to continue evading security solutions.

In this report, BlackBerry researchers examined the activities of five adversarial groups that share specific characteristics in how they are organized, operate, and in their targeting selections – a methodology that can be best described as the WINNTI approach. This ensemble, who have spent the better part of the last decade successfully targeting organizations in stealthy cross-platform attacks, continue to operate relatively undetected while undertaking multiple strategic and economic espionage operations.

The Linux Threat: This report detailed how this quintet of threat actor groups have managed to successfully infiltrate and maintain persistence on servers that comprise the backbone of the majority of large data centers using a newly identified Linux malware toolset obfuscated by a kernel-level module rootkit, all of which allows them to remain nearly undetectable on the infected systems. The fact that this new Linux malware toolset has been in the wild for the better part of the last decade without having been detected and publicly documented prior to this report makes it highly probable that the number of impacted organizations is significant and the duration of the infections lengthy.

The Windows Threat: This report also provided analysis of the use of Windows malware that attempts to elude defenders through the use of stolen adware code-signing certificates, hiding the malware in plain sight with the hopes it will be dismissed as just another blip in a nearly constant stream of adware alerts. This report contained multiple samples of the malware and the compromised code-signing certificates and recommends a new designation for malware disguised as adware that will allow defenders to better differentiate the bad from the benign and implement controls to increase detection of malware employing this tactic.

The Mobile Device Threat: This research also examined the targeting Android mobile devices by these WINNTI-related groups. A previous report from BlackBerry researchers, titled *Mobile Malware and APT Espionage: Prolific, Pervasive, and Cross-Platform* (BlackBerry, 2019), looked at APT groups increasing use of mobile malware in combination with traditional desktop malware. This report continued analysis of this tactical trend in looking at some newly discovered Android malware.

Legal Disclaimer

Appendix

Linux SHA256 Hashes

CASPER Mirai Variant:

`57cc422a6a90c571198a2d1c3db13c31fbdb48ba2f0f4356846d6d636d0f9300`

WINNTILNX Toolset:

PWNLNX1:

`0f6033d6f82ce758b576e2d8c483815e908e323d0b700040fbdab5593fb5282b`

PWNLNX2:

`08cc67002782cbafd97a4bff549d25dd72d6976d2fdf79339aaf5a3ff7c3107e`

PWNLNX3:

`08f29e234f0ce3bded1771d702f8b5963b144141727e48b8a0594f58317aac75`

PWNLNX4:

`2590ab56d46ff344f2aa4998efd1db216850bdddfc146d5d37e4b7d07c7336fc`

PWNLNX6:

`d29254ab907c9ef54349de3ec0dd8b22b4692c58ed7a7b340afbc6e44363f96a`

PWNLNX5 (Lancer Cross-Platform Controller):

`12c02b62f14cf5675e2453cbc4e884735a7c25d6288551152a0e8545b70f936a`

Lancer Network Traffic Simulation Utility:

`5455af6789342055aa04055934cca7d1873cbddf735e771130e40a9431a7c656`

Android SHA256 Hashes

PassCV Android Implant - PWNDROID4:

`ac546bd38ad2e56b42fd3e35f27048ca9c86203153868944188e6fb6822d9f63`

CASPER Android Downloaders – PWNDROID5:

`64424a7c5f0d8e1c5d64c4c6fa9bdc2987dbdcf1bafdb6f45df9e783712c5187`

Stolen Code-signing Certificates (2016-2020) and Windows Reference Samples:

Name	LivePlex Corp
Valid From	12:00 AM 04/09/2012
Valid To	11:59 PM 06/08/2014
Thumbprint	79590E622921A064FB45AB9E99D25A744BA14347
Serial Number	3F 55 42 E2 E7 1D 8D B3 57 04 1C 9D D4 5B 95 0A

Reference Samples:

```
a3fc3ca178175fa8d767d865bc983ef40ced5aaf721750c6279a1ef7faa418ac
43d66c7aad578950d8c58e4a82d32db86a67584ab09399d4c1108e7481cd92f4
36b872251991609e951aa426a24731b835a3e2a7b16f83f11ac2462439837a64
9d6677826890c037e6066ec2e25c5ca56b6c8a75b1ed70b5c68c1642800429fd
736324637ec2f43e3ec196b4674b38955de2cbf13988e269581933cf806ba8cc
7fbf5efd35ca300537949c16d9ce68b7f7b98e82bba1f95a265b8d46324d7f2c
```

Name	深圳市乐途儿科技有限公司
Valid From	11:00 PM 06/15/2015
Valid To	10:59 PM 06/15/2017
Thumbprint	37FA6C26605824B57218B1DC73BD736B458E8A48
Serial Number	68 BE C5 C0 26 4C C9 09 6D 2F B2 0A 98 86 E9 4D

(Dalek, Alexander, Crete-Nishihata, & Brooks, 2017)

Reference Samples:

```
3628efd2a0e4c28c13233dbd8353ad825865312f39cfbaff1e259f37b2dd08b5
a340af9b766b922dc0a0253784df59ca99bcaff1db33eb205faeb4c1072bdd3e
dfb39fabb3a3a8d7edb1ec3f2b90de02c5122e222a0df4260bdb6d31d898e4fe
dbd03093e58c2d60f4f47b720691cd3e6310f0566403ee0a34c2d59db9fc58d2
fc3cacb2103adedc11720c34a243de58085c1a7283ba3577b52a9fc9ab36301c
```

Name	Hangzhou Bianfeng Networking Technology Co., Ltd.
Valid From	11:00 PM 08/10/2017
Valid To	10:59 PM 08/10/2020
Thumbprint	3E2B15D5FD1CE4DF036B776CAF22244343597D34
Serial Number	0A 4E D6 BC 52 49 11 7B 35 B9 FD B7 DD 33 E8 7B

Reference Samples

```
ce3424524fd1f482a0339a3f92e440532cff97c104769837fa6ae52869013558
fbe294910ef833e1c9b2c8663c06b6ef99c13b2bc5eb01e87defb798c8066f0b
e416ad91acbc386bf67dc551fb36b9d95a195d8b656cfe4001325b8bf507624e
64cc74de6455c387218f2c09f5c1d2e149ae0c295960e9c61586c428e375ec4b
a37574387a4bacfb69e7369d6ac8749603038a1b232d9a482bbcd2dce0c091b0
266ea3df14b4a3e42ef47800ea1a70c89d184c96f9ab27059ef273176736b592
e2d2761fc2535d99527df2f7cffd8dddccd504dc0096f6d6f7fe7a4bbc032473
1ff2743e1b20f9f98e4e02dd5eb9b293e72b6dab769272c194cef11adfbfd5d0
be801280934ccb73d4ed8469ffacc9ed760ce1db25b891283c91333a15bf70d7
726ebf8743295a7facc6626a780069b3e0c82d594f8f2417a80fb679f4e7d325
1cc3978d3764c421a4ac810978f3d9e3f606c8ee7c79a7395d49b33aae16a601
```

Name	Hangzhou Shunwang Technology Co.,Ltd
Valid From	10:28 AM 12/15/2016
Valid To	10:16 AM 01/23/2018
Thumbprint	E93763AF668D36E8ACE49D299A91A0544C1CC09C
Serial Number	29 F7 33 6F 60 92 3A F0 3E 31 F2 A5

Reference Samples:

```
ef049339f1eb091cda335b51939f91e784e1ab1e006056d5a6bb526743b6cbc7
c2229a463637433451a3a50ccf3c888da8202058f5022ffd2b00fc411b395b79
1ea23560b820917b4b2d9ad8cc9cfd46d22a5bed5356702e9edb699bae1c0e5d
d07af16c19a467fbac0a5173b0aa4c4a85863335ac9bb3f60d1bf2638b7ccc7d
8674c76583c13c60fcb6dc344bae4a5149cce35a85bb600f0a6af5e769b98585
944769de07f8599fbf4ec1651900d119d5896c85d8aabd694922ac71ebd4fd6
```

Name	Xiamen Tongbu Networks Ltd.
Valid From	11:00 PM 04/27/2016
Valid To	10:59 PM 06/27/2019
Thumbprint	B5DBB22B4F3EDE0B6A9987ADEB71C0E67CC30798
Serial Number	32 D3 8A BD DE 43 8F 81 72 97 BE 45 8E FB 4C D4

Reference Samples:

```
9b7c1e37d5f56cc0b5e5e22ce9805e237a189297e78405b9c392a0953b6e0321
42eab05c611bf24d86bb6c985caa2ad7380ed7d98340c7f08de9361be14dc244
```

Name	Elex do Brasil Participações Ltda
Valid From	1:00 AM 4/13/2015
Valid To	12:59 AM 7/13/2017
Thumbprint	C94CE34C0B799BA99CD97620FA14EF5A91F98931
Serial Number	06 71 EE 52 6A CB 6F 9B E2 01 F5 A8 E2 03 C4 1C

Reference Samples:

```
83125b051e1f31051e58041597573ab8743c81cce61b4da8025a1cfcff4e6e80
af30d617dd0edb4f4107457674951cec28a276215c92b8fc64112ccdbbd32445
fe61dc240c8854614bc57f0ef5a4ffcaf3852a4c9d64d759bed41f990f7dcc99
0d132fddc55941caeca2b2777cd555ebac728a6e0fcc3fe3a07d4a6376f57691
e1be51b7e59518bcae7232291fda614033eba56e8cb4578dcbf721f80bb8da37
```

Name	Shanghai Hintsoft Co., Ltd.
Valid From	06:17 AM 11/02/2016
Valid To	04:29 AM 08/27/2019
Thumbprint	98549AE51B7208BDA60B7309B415D887C385864B
Serial Number	09 89 C9 78 04 C9 3E C0 00 4E 28 43

Reference Samples:

```
28924b6329f5410a5cca30f3530a3fb8a97c23c9509a192f2092cbdf139a91d8
b28c024db80cf3e7d5b24ccc9342014de19be990efe154ba9a7d17d9e158eecb
```

Name	2345.com
Valid From	12:00 AM 02/23/2016
Valid To	11:59 PM 09/02/2017
Thumbprint	741C8C6560CFF170368F3F73CA4D03C853C78F8B
Serial Number	21 23 96 86 F6 6C B0 BC 11 21 E5 78 87 23 62 8A

Reference Samples:

```
1ff2743e1b20f9f98e4e02dd5eb9b293e72b6dab769272c194cef11adfbfd5d0
2f4b48457d8465347d1d40b040fa246f3b8b657531304238231c8b1e92100e78
65d21c3374e332e2bfeedd3ec7ab0df67b57b676dd2d52a2e2c389f844aa7a18
```

Name	Polypower Technology Co., Limited
Valid From	10:02 AM 5/28/2015
Valid To	10:02 AM 6/27/2016
Thumbprint	01ED0A76185E76575F8FCA667DA73AD290656E03
Serial Number	11 21 A3 9E 97 47 48 62 3C A6 E3 E4 9A 8B AE B3 ED 3A

Reference Samples:

```
57be4485c43dc461b4a8f43fb7fb0d7a4550da130148f8634dca88bd9366de53
7929af1c8e1c1c575f807b617e60586393bd3be1922cc4541fdd69975f90fc5b
```

Name	SDL PLC
Valid From	11:00 PM 08/06/2014
Valid To	12:00 PM 11/30/2017
Thumbprint	73563226C7BE012BAF171FE91BBFDC18190A96D9
Serial Number	06 42 A9 8B CD A2 D2 15 83 F1 FF 5F A5 0C 94 41

Reference Samples:

```
503d9e4be006218902c5eeada66f2bf76c6efb0cb5d06300fc9246dda668007a
71f188e26d6ecda3462da3bfa81b956de71e05fd045a7f66d0b5528a9d7aca36
```

Works Cited

Akamai. (2015, September 29). *XOR DDoS Botnet Launching 20 Attacks A Day From Compromised Linux Machines.* Retrieved from Akamai: https://www.akamai.com/us/en/about/news/press/2015-press/xor-ddos-botnet-attacking-linux-machines.jsp

Akamai. (2015, November 12). *Case Study: FastDNS Infrastructure battles Xor Botnet.* Retrieved from Akamai: https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/fast-dns-xor-botnet-case-study.pdf

Android. (2020, January 6). *Codenames, Tags, and Build Numbers. Retrieved from* Android Source: https://source.android.com/setup/start/build-numbers

Avast Threat Intelligence Team. (2015, January 6). *Linux DDoS Trojan hiding itself with an embedded rootkit.* Retrieved from Avast Blog: https://blog.avast.com/2015/01/06/linux-ddos-trojan-hiding-itself-with-an-embedded-rootkit/

Baumgartner, K., & Raiu, C. (2014, December 8). *The 'Penguin' Turla.* Retrieved from Kaspersky Lab: https://securelist.com/the-penquin-turla-2/67962/

BlackBerry (2019, October 24). *Mobile Malware and APT Espionage: Prolific, Pervasive, and Cross-Platform.* Retrieved from ThreatVector Blog: https://threatvector.cylance.com/en_us/home/mobile-malware-and-apt-espionage-prolific-pervasive-and-cross-platform.html

Bradley, J. (2018, July 28). *MACDOORED: A First Look Into Real-World MACOS Intrustions.* Retrieved from Shakacon: https://objectivebythesea.com/v1/talks/OBTS_v1_Bradley.pdf

Chronicle. (2019, May 15). Winnti: More than just Windows and Gates. Retrieved from Chronicle Blog: https://medium.com/chronicle-blog/winnti-more-than-just-windows-and-gates-e4f03436031a

Command Five Pty Ltd. (2011, September 6). *SK Hack by an Advanced Persistent Threat.* Retrieved from Kaspersky CDN: https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2013/04/20082912/C5_APT_SKHack.pdf

Coppola, M. (2013, October 11). *An LKM rootkit targeting Linux 2.6/3.x on x86(_64), and ARM.* Retrieved from Github: https://github.com/mncoppola/suterusu

Coppola, M. (2013, January 7). *Suterusu Rootkit: Inline Kernel Function Hooking on x86 and ARM.* Retrieved from Michael Coppola's Blog: https://poppopret.org/2013/01/07/suterusu-rootkit-inline-kernel-function-hooking-on-x86-and-arm/

Cylance Threat Research Team. (2016, October 18). *Digitally Signed Malware Targeting Gaming Companies.* Retrieved from Cylance Threat Vector: https://threatvector.cylance.com/en_us/home/digitally-signed-malware-targeting-gaming-companies.html

Dalek, J., Alexander, G., Crete-Nishihata, M., & Brooks, M. (2017, July 5). *Insider Information: An intrusion campaign targeting Chinese language news sites.* Retrieved from Citizen Lab: https://citizenlab.ca/2017/07/insider-information-an-intrusion-campaign-targeting-chinese-language-news-sites

Damari, O. (2019, November 1). *Autochk Rootkit Analysis.* Retrieved from Low Level Pleasure: https://repnz.github.io/posts/autochk-rootkit-analysis/

Dell Counter Threat Unit Research Team. (2019, February 27). *A Peek into BRONZE UNION's Toolbox.* Retrieved from Secureworks: https://www.secureworks.com/research/a-peek-into-bronze-unions-toolbox

Department of Justice (2018, November 1). *Attorney General Jeff Sessions Announces New Initiative to Combat Chinese Economic Espionage.* Retrieved from: https://www.justice.gov/opa/speech/attorney-general-jeff-sessions-announces-new-initiative-combat-chinese-economic-espionage

Department of Justice (2020, Febuary 6). *Attorney General William P. Barr Delivers the Keynote Address at the Department of Justice's China Initiative Conference.* Retrieved from: https://www.justice.gov/opa/speech/attorney-general-william-p-barr-delivers-keynote-address-department-justices-china

Fraser, N., Plan, F., O'Leary, J., Cannon, V., Leong, R., Perez, D., & Shen, C.-e. (2019, August 7). *Double Dragon APT41: a dual espionage and cyber crime operation.* Retrieved from FireEye Threat Research: https://content.fireeye.com/apt41/rpt-apt41

Gonzalez, R. (2019, October 2). *The Emissary Panda.* Retrieved from Medium: https://medium.com/@clermont1050/the-emissary-panda-ba6876e28d4b

Hannas, W. C., Mulvenon, J., & Puglisi, A. B. (2013). *Chinese Industrial Espionage: Technology Acquisition and Military Modernization.* London and New York: Routledge.

Hegel, T. (2018, May 3). Burning Umbrella: *An Intelligence Report on the Winnti Umbrella and Associated State-Sponsored Attackers.* Retrieved from 401trg: https://401trg.com/burning-umbrella/

idlefire. (2016, December 30). 内网穿透(跨平台). Retrieved from GitHub: https://github.com/idlefire/ew

Jang, I. (2019, February 7). *Implementing a New Custom Netlink Family Protocol.* Retrieved from Better Tomorrow With Computer Science: https://insujang.github.io/2019-02-07/implementing-a-new-custom-netlink-family-protocol/

Kaspersky Lab Global Research and Analysis Team. (2013, April). *Winnti: More Than Just a Game.* Retrieved from https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/20134508/winnti-more-than-just-a-game-130410.pdf

Krebs, B. (2017, December 13). *Mirai IoT Botnet Co-Authors Plead Guilty.* Retrieved from Krebs on Security: https://krebsonsecurity.com/2017/12/mirai-iot-botnet-co-authors-plead-guilty/

Linux Foundation. (2020). *Linux - The Linux Foundation*. Retrieved from The Linux Foundation: https://www.linuxfoundation.org/projects/linux/

Malware Must Die! (2014, September 29). *MMD-0028-2014 - Linux/XOR.DDoS : Fuzzy reversing a new China ELF.* Retrieved from Malware Must Die! Blog: https://blog.malwaremustdie.org/2014/09/mmd-0028-2014-fuzzy-reversing-new-china.html

MITRE. (2017, May 31). *MITRE ATT&CK*. Retrieved from MITRE: https://attack.mitre.org/software/S0021/

National Cyber Security Centre. (2018, October 11). *Joint Report on Publicly Available Hacking Tools.* Retrieved from National Cyber Security Centre: https://www.ncsc.govt.nz/assets/NCSC-Documents/GSA-2018-133-Joint-report-on-publicly-available-hacking-tools.pdf

Netcraft. (2019, August). *August 2019 Web Server Survey*. Retrieved from Netcraft: https://news.netcraft.com/archives/category/web-server-survey/

PrudentWoo. (2014, December 31). *gzexe* 助shell脚本加密 01. Retrieved from CSDN: https://blog.csdn.net/wuweilong/article/details/42290839

Red Hat. (2019, November 6). *Red Hat Enterprise Linux Release Dates.* Retrieved from Red Hat Customer Portal: https://access.redhat.com/articles/3078

Rootkiter. (2015, May 12). *EarthWorm.* Retrieved from GitHub: https://github.com/rootkiter/EarthWorm

SDT. (2019, January 4). *Shadowsocks: A secure SOCKS5 proxy.* Retrieved from Shadowsocks: https://shadowsocks.org/assets/whitepaper.pdf

Sklyarov, I. (2007, January 1). Retrieved from Google Books: https://books.google.com/books?id=yqHVAwAAQBAJ&pg=PA317&lpg=PA317&dq=%22orig_readdir%22&source=bl&ots=r1NI0vpGCA&sig=ACfU3U0vpQbluQZMIQqiYBWAgGgspeMbpQ&hl=en&sa=X&ved=2ahUKEwjyhYzJ2P7mAhXJfFAKHd-IDcYQ6AEwAXoECAoQAQ#v=onepage&q=%22orig_readdir%22&f=false

SophosLabs. (2017, May 2). *Super Free Music Player in Google Play is malware: a technical analysis*. Retrieved from Naked Security by Sophos: https://nakedsecurity.sophos.com/2017/05/02/super-free-music-player-in-google-play-is-malware-a-technical-analysis/

Symantec. (2011, October 27). *Backdoor.Winnti*. Retrieved from Symantec Security Center: https://web.archive.org/web/20190410223403/https://www.symantec.com/security-center/writeup/2011-102716-2809-99

Symantec Threat Intelligence. (2018, February 27). *Chafer: Latest Attacks Reveal Heightened Ambitions*. Retrieved from Symantec Blog: https://www.symantec.com/blogs/threat-intelligence/chafer-latest-attacks-reveal-heightened-ambitions

The Internet Archive. (2020, March 2). Retrieved from Wayback Machine: https://web.archive.org/web/20200302190941/https://www.worldwiredlabs.com/android-support/

United States of America v. Zhang et al, 13CR3132-H (United States District Court, Southern District of California June 2017).

Wikipedia. (2020, January 5). *Broiler*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Broiler

World Wired Labs. (2020, January 29). *World Wired Labs Pricing*. Retrieved from World Wired Labs: https://www.worldwiredlabs.com/_pricing_/

BlackBerry®

Intelligent Security. Everywhere.