Analysis of a Recent PlugX Variant - “P2P PlugX”

This is Shusei Tomonaga at Analysis Center.

PlugX, a Remote Access Tool (RAT) often seen in many APT cases, has been in the wild for some years. Various sectors in Japan have been suffering from this type of attack from 2012, and Analysis Center has been working to catch up on the evolution of the PlugX family since then.

In this blog post, I will write about a recent PlugX variant which we first encountered in October 2014. The variant has interesting new aspects and the most significant one, in my view, is the P2P function - so let me tentatively name it “P2P PlugX”.

Size Expansion of Configuration Information

PlugX is designed to run based on its configuration information stored in itself. Our analysis revealed that the size of the configuration information has been expanded for the recent variant. While the former ones have either 0x2540 bytes (Observed since August 2013) or 0x2d58 bytes (Observed since June 2014), the recent one has 0x36a4 bytes, roughly 20% larger in size. This has led it to do more, such as:

- Communication with more C&C servers – up to 16
- P2P communication between infected nodes
- MAC address check - PlugX runs if the MAC address of an infected host coincides with configuration information in itself (If not specified in the configuration, PlugX runs on any host).
- (To bypass UAC) configurable setting for the process to abuse

Other than these, new coding algorithm has been introduced.

I will pick up some of the interesting features for more description. For details of the configuration file, you can refer to Appendix A in the bottom of this post.

Additional Communication Protocol for C&C Servers

Former versions of PlugX used to set four C&C Server addresses to communicate with. With the P2P PlugX, attackers can set up to 16 C&C servers. Communication protocol with C&C servers has also been improved.

Former PlugX could only configure four communication protocols, but for P2P PlugX, protocol number 255 became available. This protocol is reserved by IANA, but no specific application is assigned.

Table 1: Configurations and Communication Protocols which

Table 1: Configurations and Communication Protocol which PlugX uses to connect to C&C Servers

Configuration No.	Protocol Number (In IP header)	Data Format
1	6 (TCP)	Binary
2	6 (TCP)	HTTP
3	17 (UDP)	DNS
4	1 (ICMP)	Binary
5	255	Binary

P2P Function Enabled

P2P PlugX can communicate with other similarly-infected hosts. When one PlugX succeeds to infect a host, it then accesses to every IP address in the local network one-by-one and communicate with any connectable nodes, using one of the following protocols listed in Table 2.

Table 2: Configurations and Communication Protocols which P2P PlugX uses to communicate by P2P

Configuration No.	Protocol Number (In IP header)	Data Format
1	6 (TCP)	Binary
2	17 (UDP)	DNS
3	1 (ICMP)	Binary
4	255	Binary

With P2P protocol, even if a PlugX exists in an environment with no direct access to the Internet, it may communicate with C&C server through other infected hosts. We have also seen some P2P-disabled samples.

Note that this P2P communication theoretically can be applied to any other TCP/UDP ports. But in cases which JPCERT/CC has observed, P2P PlugX only uses either TCP/1357 or UDP/1357 for P2P communication. If you see any scanning activity to TCP/1357 or UDP/1357, we highly recommend that you conduct further investigation.

New Encoding Algorithm

PlugX uses a single encoding algorithm for inbound/outbound data, configuration, key logging data and strings used internally. Its encoding method has been modified from time to time, aligned with major upgrade of PlugX itself.

Likewise, P2P PlugX has a new encoding algorithm. Here's a python code to decode.

```
def plugx_decode(data):
    decode_key = struct.unpack_from('<I', data, 0)[0]
    out = ''

    # XOR Values might possibly be varied.
    key1 = decode_key ^ 20140918
    key2 = decode_key ^ 353

    for c in data[4:]:
        # ADD/SUB Values might possibly be varied.
        key1 += 3373
        key2 -= 39779

        dec = int(c) ^ (((key2 >> 16) & 0xff ^ ((key2 & 0xff ^ (((key1 >> 16) & 0xff ^ (key1 - (key1 >> 8) & 0xff)) - (key1 >> 24) & 0xff)) - (key2 >> 8) & 0xff)) - (key2 >> 24) & 0xff)
        out = out + chr(dec)

    return out
```

What's Next?

P2P PlugX introduced several new features which surely made attackers to manage their attack infrastructure efficiently. We are sure that PlugX will keep evolving, and continuous analysis will be necessary for preventing/mitigating possible incident. We will keep you updated on any new findings.

Thank you very much for reading.

- Shusei Tomonaga

(For any inquiry or incident report regarding PlugX, please contact info[at]jpcert.or.jp)

Appendix A: Entire Configuration of P2P PlugX

Table 3: Entire Configuration of P2P PlugX

Offset	Length	Description
0x0000	20	Not used
0x0014	4	Flag if remove own DLL from list of modules
0x0018	4	Flag enable/disable key logger
0x001c	12	Not used

0x0028	4	Duration of suspend activity
0x002c	4	Duration of suspend activity
0x0030	672	Network Access Flag (for a week with 15min interval)
0x02d0	4 * 4	DNS Server IP Address x 4
0x02e0	68 * 16	control Server Information x 16
0x0720	128 * 16	HTTP Access URL x 16
0x0f20	196 * 4	Proxy/authentication config x 4
0x1230	4	Method to make it resident (e.g. Create Service. Create Run Key)
0x1234	512	Folder to Install
0x1434	512	Service Name
0x1634	512	Service Display Name
0x1834	512	Service Description
0x1a34	4	Registry Root Key Value for Run Registry Key Configuration
0x1a38	512	Run Registry Key Name
0x1c38	512	Run Registry Key Value
0x1e38	4	Enable/Disable Code injection
0x1e3c	512 * 4	Program Name for Code Injection x 4
0x263c	4	Enable/Disable Code injection for UAC Bypass
0x2640	512 * 4	Program Name to inject code for UAC Bypass x 4
0x2e40	512	Authentication Character String for PlugX
0x3040	512	Authentication Character String for C&C Server
0x3240	512	Mutex Name
0x3440	4	Enable/Disable Screen Capture
0x3444	4 * 5	Screen Capture Configuration Value
0x3458	528	Folder to Store Screen Captures
0x3658	4	Enable/Disable P2P(TCP)
0x365c	4	P2P(TCP) Port Number
0x3660	4	Enable/Disable P2P(UDP)
0x3664	4	P2P(UDP) Port Number
0x3668	4	Enable/Disable P2P(ICMP)
0x366c	4	P2P(ICMP) Port Number
0x3670	4	Enable/Disable P2P(IP Protocol Number 255)
0x3674	4	P2P(IP Protocol Number 255) Port Number
0x3678	4	Enable/Disable P2P Scanning
0x367c	4 * 4	P2P Scanning Beginning Address x 4
0x368c	4 * 4	P2P Scanning End Address x 4
0x369c	6	Run program if this MAC Address is used
0x36a2	2	Not used

Appendix B: SHA-256 hash value of P2P PlugX

bc65e2859f243ff45b12cd184bfed7b809f74e67e5bb61bc92ed94058d3d2515

93c85a8dd0becc4e396eea2dc15c0010ff58d2b873d44fd7e45711a27cfe613b

0ff134057a8b2e31b148fedfdd185f5b1a512149499a8c5c0915cf10b10a613e