A Look Into Fysbis: Sofacy's Linux Backdoor

posted by: Bryan Lee and Rob Downs on February 12, 2016 3:00 PM
filed in: Malware, Threat Prevention, Unit 42
tagged: Fysbis, Linux, Sofacy

Introduction

The Sofacy group, also known as APT28 and Sednit, is a fairly well known cyber espionage group believed to have ties to Russia. Their targets have spanned all across the world, with a focus on government, defense organizations and various Eastern European governments. There have been numerous reports on their activities, to the extent that a Wikipedia entry has even been created for them.

From these reports, we know that the group uses an abundance of tools and tactics, ranging across zero-day exploits targeting common applications such as Java or Microsoft Office, heavy use of spear-phishing attacks, compromising legitimate websites to stage watering-hole attacks, and targeting over a variety of operating systems – Windows, OSX, Linux, even mobile iOS.

The Linux malware Fysbis is a preferred tool of Sofacy, and though it is not particularly sophisticated, Linux security in general is still a maturing area, especially in regards to malware. In short, it is entirely plausible that this tool has contributed to the success of associated attacks by this group. This blog post focuses specifically on this Linux tool preferred by Sofacy and describes considerations and implications when it comes to Linux malware.

Malware Assessment

Fysbis is a modular Linux trojan / backdoor that implements plug-in and controller modules as distinct classes. For reference, some vendors categorize this malware under the Sednit attacker group naming designation. This malware includes both 32-bit and 64-bit versions of Executable and Linking Format (ELF) binaries. Additionally, Fysbis can install itself to a victim system with or without root privileges. This increases the options available to an adversary when it comes to selecting accounts for installation.

Summary information for the three binaries we analyzed follows:

MD5	364ff454dcf00420cff13a57bcb78467
SHA-256	8bca0031f3b691421cb15f9c6e71ce19335 5d2d8cf2b190438b6962761d0c6bb
ssdeep	3072:n+1R4tREtGN4qyGCXdHPYK9l0H786 O26BmMAwyWMn/qwwiHNl:n+1R43QcIL XdF0w6IBmMAwwCwwi
Size	141.2 KB (144560 bytes)
Type	ELF 64-bit (stripped)
Install as root	/bin/rsyncd

Root install desc	synchronize and backup service
Install as non-root	~/.config/dbus-notifier/dbus-inotifier
Non-root install desc	system service d-bus notifier
C2	azureon-line[.]com (TCP/80)
Usage Timeframe	Late 2014

Table 1: Sample 1 – Late 2014 Sofacy 64-bit Fysbis

MD5	075b6695ab63f36af65f7ffd45cccd39
SHA-256	02c7cf55fd5c5809ce2dce56085ba43795f2 480423a4256537bfdfda0df85592
ssdeep	3072:9ZAxHANuat3WWFY9nqjwbuZf454U NqRpROIDLHaSeWb3LGmPTrIW33HxIajF: 9ZAxHANJAvbuZf454UN+rv eQLZPTrV3Z
Size	175.9 KB (180148 bytes)
Type	ELF 32-bit (stripped)
Install as root	/bin/ksysdefd
Root install desc	system kernel service defender
Install as non-root	~/.config/ksysdef/ksysdefd
Non-root install desc	system kernel service defender
C2	198.105.125[.]74 (TCP/80)
Usage Timeframe	Early 2015

Table 2: Sample 2 – Early 2015 Sofacy 32-bit Fysbis

MD5	e107c5c84ded6cd9391aede7f04d64c8
SHA-256	fd8b2ea9a2e8a67e4cb3904b49c789d57ed 9b1ce5bebfe54fe3d98214d6a0f61
ssdeep	6144:W/D5tpLWtr91gmaVy+mdckn6BCUd c4mLc2B9:4D5Lqgkcj+
Size	314.4 KB (321902 bytes)
Type	ELF 64-bit (not stripped)
Install as root	/bin/ksysdefd
Root install desc	system kernel service defender
Install as non-root	~/.config/ksysdef/ksysdefd
Non-root install desc	system kernel service defender
C2	mozilla-plugins[.]com (TCP/80)
Usage Timeframe	Late 2015

Table 3: Sample 3 – Late 2015 Sofacy 64-bit Fysbis

Overall, these binaries are assessed as low sophistication, but effective. They epitomize the grudging reality that Advanced Persistent Threat (APT) actors often don’t require advanced means to affect their objectives. Rather, these actors more often than not hold their advanced malware and zero day exploits in reserve and employ just enough resources to meet their goals. It is only fair that defenders use any shortcuts or tricks at their disposal to shorten the amount of time it takes to assess threats. In other words, defenders should always look for ways to work smarter before they have to work harder.

Getting the Most Out of Strings

Binary strings alone revealed a good amount about these files, increasing the efficacy of activities such as static analysis categorization (e.g., Yara). One example of this is Fysbis installation and platform targeting information for the samples in Table 1 and Table 2.

Figure 1: Sofacy Fysbis installation and platform targeting found in strings

In this case, we can see the binary installation path and local reconnaissance to determine which flavor of Linux the malware is running. This is followed by a number of Linux shell command style commands related to the malware establishing persistence.

Another example of easily obtained information from these samples is capability based.

```
<font size=4 color=red align=center>WRITE FILE IS SUCCESS</font><br>
<font size=4 color=red align=center>WRITE FILE IS NOT SUCCESS</font><br>
<table><caption><font size=4 color=red>TABLE FIND FILES</font></caption>
</table>
<table><caption><font size=4 color=red>TABLE READ FILES</font></caption>
<table><caption><font size=4 color=red>TABLE DELETE FILES</font></caption>
<table><caption><font size=4 color=red>TABLE EXECUTE FILES</font></caption>
8FSModule
/bin/sh
Your command not writed to pipe
Success execute command or long for waiting executing your command
Terminal started
Terminal don`t started
Terminal stopped
Terminal don`t stopped
Terminal yet started
Terminal yet stopped
Incorrect command ID
Error for command ID
Terminal don`t started for executing command
Command will have end with \n
exit
11RemoteShell
w -u 2>&1
</pre></font><pre style="font-weight:bold;">
</pre></font><pre>Keylogger started
</pre>
</pre></font><pre>Keylogger not started
</pre>
</pre></font><pre>Keylogger stoped
</pre>
</pre></font><pre>Keylogger not stoped
</pre>
</pre></font><pre>Keylog thread exit
</pre>
</pre></font><pre>Keylogger yet started
</pre>
```

Figure 2: Sofacy Fysbis capability related leakage through strings

Figure 2 shows interactive status / feedback strings that can give a defender an initial profile of capabilities. In addition to contributing to static analysis detections, this can be useful as a starting point for further incident response prioritization and qualification of the threat.

Symbolic Information Can Shorten Analysis Time

Interestingly, the most recent ELF 64-bit binary we analyzed (Table 3) was not stripped prior to delivery, which offered additional context in the form of symbolic information. Defenders more familiar with Windows Portable Executable (PE) binaries can equate this with compilation of a Debug version versus a Release version. For comparison, if we were to inspect Fysbis "RemoteShell" associated strings in one of the stripped variants, we would only see the following:

```
14:11 $ strings 02c7cf55fd5c5809ce2dce56085ba43795f2480423a4256537bfdfda0df85592.bin | grep "RemoteShell"
11RemoteShell
```

Figure 3: Sofacy Fysbis stripped binary string references to RemoteShell capability

Compare this with what is available from the non-stripped variant:

```
14:13 $ strings fd8b2ea9a2e8a67e4cb3904b49c789d57ed9b1ce5bebfe54fe3d98214d6a0f61.bin | grep "RemoteShell"
11RemoteShell
_ZTV11RemoteShell
RemoteShell.cpp
_GLOBAL__I__ZN11RemoteShellC2Ev
_ZN11RemoteShellD1Ev
_ZN11RemoteShellC2Ev
_ZN11RemoteShell20kCGkHUEgaGDhXNYMwROTEv
_ZN11RemoteShell14executeCommandEhPhi
_ZN11RemoteShellC1Ev
_ZN11RemoteShell13startTerminalEv
_ZN11RemoteShell20ESPcFfSyPjDQJZKbRqttEv
_ZTS11RemoteShell
_ZTV11RemoteShell
_ZN11RemoteShell20bKeKOVTxVflxGpNXwNlXENSt3tr110shared_ptrI20AjcgRSsaDPvrGegGDWvtEE
_ZN11RemoteShell16readIntoListLogsEPhj
_ZTI11RemoteShell
_ZN11RemoteShell17terminateTerminalEv
_ZN11RemoteShell7sendLogEi
_ZN11RemoteShellD2Ev
_ZN11RemoteShell11executeBashEPhi
```

Figure 4: Sofacy Fysbis non-stripped binary strings referenes to RemoteShell capability

Little static analysis gifts like these can help to speed defender enumeration of capabilities and – more importantly – further contribute to correlation and detection across related samples.

Additionally, this latest sample demonstrated minor evolution of the threat, most notably in terms of obfuscation. Specifically, both samples in Table 1 and Table 2 leaked installation information in the clear within binary strings. This was not the case with the sample in Table 3. Taking a closer look at this non-stripped binary using a disassembler, the following corresponds to decoding malware installation information for a root-privilege account.

```
mov     edx, 0Bh            ; unsigned __int64
lea     rsi, _ZN15installrootvars17INSTALL_ROOT_MASKE ; unsigned __int8 *
mov     rdi, rbp            ; this
call    _ZN20vUcsxhhCQAcpKxdqawQx20RzYjYPwTtIFdZPcbENJoEPhm ; vUcsxhhCQAcpKxdqawQx::RzYjYPwTtIFdZPcbENJo(uchar *,ulong)
mov     edx, 5              ; unsigned __int64
lea     rsi, _ZN15installrootvars23INSTALLROOT_XAGENT_PATHE ; unsigned __int8 *
mov     rdi, rbp            ; this
call    _ZN20vUcsxhhCQAcpKxdqawQx20ubXdkibSUiOOMbkwPaloEPhm ; vUcsxhhCQAcpKxdqawQx::ubXdkibSUiOOMbkwPalo(uchar *,ulong)
lea     r12, [rsp+188h+var_178]
mov     rsi, rbp
mov     rdi, r12            ; this
call    _ZN20vUcsxhhCQAcpKxdqawQx20FMUUcPZMVvfdaECNazdbEv ; vUcsxhhCQAcpKxdqawQx::FMUUcPZMVvfdaECNazdb(void)
mov     edx, 9              ; unsigned __int64
lea     rsi, _ZN15installrootvars23INSTALLROOT_XAGENT_NAMEE ; unsigned __int8 *
mov     rdi, rbp            ; this
call    _ZN20vUcsxhhCQAcpKxdqawQx20ubXdkibSUiOOMbkwPaloEPhm ; vUcsxhhCQAcpKxdqawQx::ubXdkibSUiOOMbkwPalo(uchar *,ulong)
```

Figure 5: Assembly code view of Sample 3 installation decoding

In this case, the symbolic information hints at the method used for decoding, with references to mask, path, name, and info byte arrays.

```
.data:00000000006300EF                 db    0
.data:00000000006300F0 ; unsigned __int8 installrootvars::INSTALL_ROOT_MASK
.data:00000000006300F0 _ZN15installrootvars17INSTALL_ROOT_MASKE db 0FAh
.data:00000000006300F0                                         ; DATA XREF: main+67↑o
.data:00000000006300F1                 db 0ADh ; �
.data:00000000006300F2                 db 0EEh ; �
.data:00000000006300F3                 db  14h
.data:00000000006300F4                 db  78h ; x
.data:00000000006300F5                 db  52h ; R
.data:00000000006300F6                 db  54h ; T
.data:00000000006300F7                 db  78h ; x
.data:00000000006300F8                 db 0A5h ; �
.data:00000000006300F9                 db 0A8h ; �
.data:00000000006300FA                 db  14h
.data:00000000006300FB ; unsigned __int8 installrootvars::INSTALLROOT_XAGENT_PATH
.data:00000000006300FB _ZN15installrootvars23INSTALLROOT_XAGENT_PATHE db 0D5h
.data:00000000006300FB                                         ; DATA XREF: main+7B↑o
.data:00000000006300FC                 db  1Ah
.data:00000000006300FD                 db  9Dh ; �
.data:00000000006300FE                 db 0E7h ; �
.data:00000000006300FF                 db  9Fh ; �
.data:0000000000630100 ; unsigned __int8 installrootvars::INSTALLROOT_XAGENT_NAME
.data:0000000000630100 _ZN15installrootvars23INSTALLROOT_XAGENT_NAMEE db 91h
.data:0000000000630100                                         ; DATA XREF: main+9F↑o
.data:0000000000630101                 db  4Fh ; O
.data:0000000000630102                 db 0D8h ; �
.data:0000000000630103                 db 0BFh ; �
.data:0000000000630104                 db 0A3h ; �
.data:0000000000630105                 db  94h ; �
.data:0000000000630106                 db 0A6h ; �
.data:0000000000630107                 db 0BAh ; �
.data:0000000000630108                 db  1Fh
.data:0000000000630109                 db    0
.data:000000000063010A                 db    0
.data:000000000063010B                 db    0
.data:000000000063010C                 db    0
.data:000000000063010D                 db    0
.data:000000000063010E                 db    0
.data:000000000063010F                 db    0
```

Figure 6: Assembly view of Sample 3 root installation related byte arrays

As it turns out, the referenced byte mask is applied to the other byte arrays using a rolling double-XOR algorithm to construct malware installation paths, filenames, and descriptions for a Linux root account. Corresponding INSTALLUSER byte arrays exist, which facilitate the non-root installation for the trojan. The same masking method is also used by the binary to decode malware

configuration C2 information, further showcasing how a little symbolic information can go a long way towards completeness and higher confidence in assessment of a malware sample.
If you would like to learn more about how Fysbis works, the samples analyzed remain fairly consistent with the sample analysis found here.

Infrastructure Analysis

As Unit 42 has discussed in depth in other blog articles, we have observed that adversaries in general are seemingly hesitant in changing their infrastructure. This may be due to not wanting to commit additional resources, or simply a matter of retaining familiarity for the sake of timeliness. In either case, we see the same type of behavior here with the Fysbis samples in use by Sofacy.
The oldest sample (Table 1), was found to beacon to the domain azureon-line[.]com, which had already been widely publicized as a known command and control domain for the Sofacy group. Using passive DNS, we can see that two of the original IPs this domain resolved to, 193.169.244[.]190 and 111.90.148[.]148 also mapped to a number of other domains that had been in use by the Sofacy group during that time period.

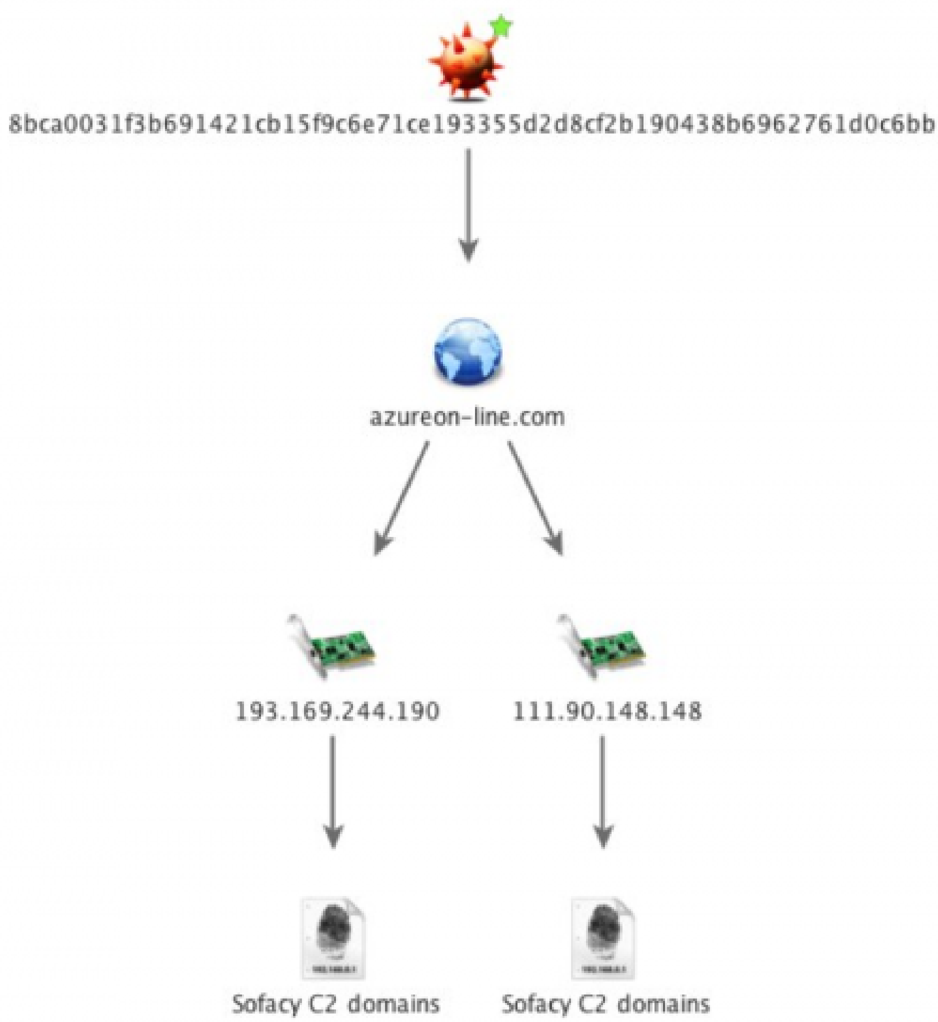

Figure 7: Sample 1 C2 resolutions

The first of the newer samples (Table 2), continues the trend and beacons to an IP also widely associated with the Sofacy group, 198.105.125[.]74. This IP has been mostly associated with the tool specifically known as CHOPSTICK, which can be read about here.

Figure 8: Sample 2 C2 resolutions

The newest sample (Table 3), introduces a previously unknown command and control beacon to mozilla-plugins[.]com. This activity aligns with the previously observed Sofacy group tactic of integrating legitimate company references into their infrastructure naming convention. Neither this new domain nor the IP it resolves to have been observed in the past, indicating that the sample in Table 3 may be associated with a newer campaign. Comparing this sample's binary with the other two however, shows there are significant similarities on the code level as well as in terms of shared behavior.

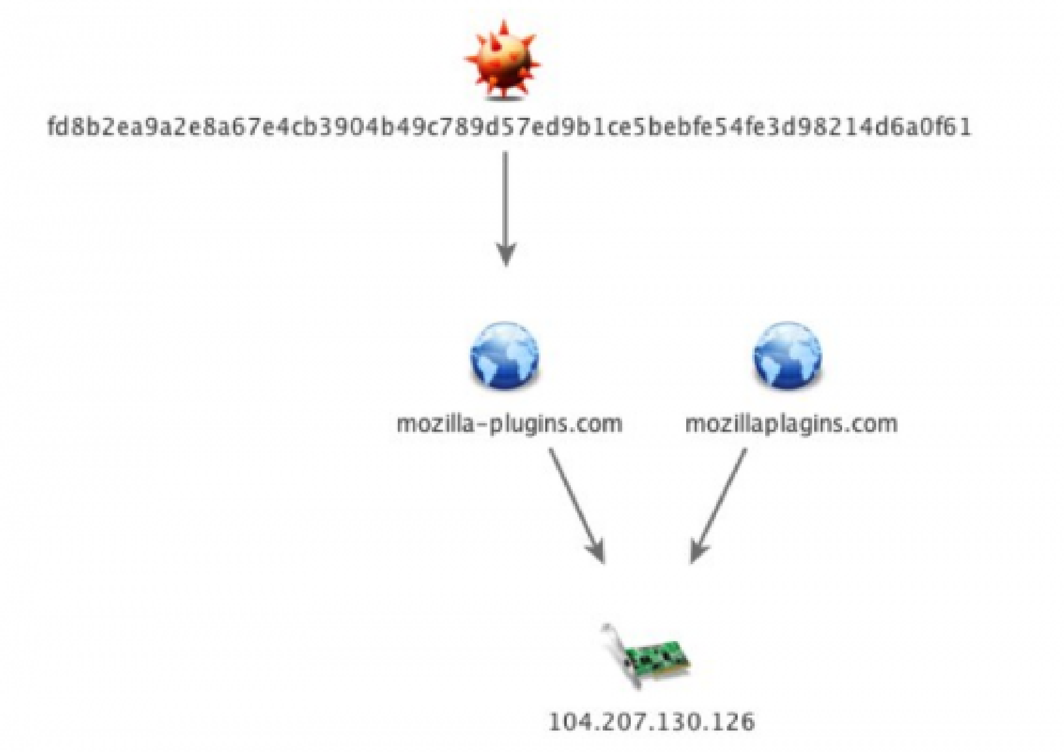

Figure 9: Sample 3 C2 resolutions

Conclusion

Linux is used across business and home environments and appears in a variety of form factors. It is a preferred platform within data centers and the cloud for businesses, as well as an ongoing favorite when it comes to a majority of Internet-facing web and application servers. Linux is also at the foundation of Android devices and a number of other embedded systems. The value proposition of Linux – especially when it comes to its use in the enterprise – can be broken out into three perceived benefits: lower total cost of ownership (TCO), security, and feature set. While numbers and comparison alone can contribute to measurement of TCO and feature set, security requires further qualification. Expertise in the Linux platform is highly sought after across all industries for multiple disciplines, from system administration to big data analytics to incident response.

The majority of businesses still maintain Windows-heavy user environments where certain core infrastructure components also operate under Windows servers (e.g., Active Directory, SharePoint, etc.). This means, from a practical perspective, most of a business's focus remains on supporting and protecting Windows assets. Linux remains a mystery to a number of enterprise IT specialists –most critically for network defenders. Identifying and qualifying potential incidents requires a familiarity with what constitutes normal operation in order to isolate anomalies. The same is true for any other asset in an environment, normal operation is entirely dependent on a given asset's role / function in the enterprise.

Lack of expertise and visibility into non-Windows platforms combine in some environments to present significant risks against an organization's security posture. As a recent caution, the Linux vulnerability described under CVE-2016-0728 further demonstrates the potential breadth of real-world risks to associated platforms. A natural extension of this exposure is increased targeting by both dedicated and opportunistic attackers across various malicious actor motivations. Despite the lingering belief (and false sense of security) that Linux inherently yields higher degrees of protection from malicious actors, Linux malware and vulnerabilities do exist and are in use by advanced adversaries. To mitigate associated risks requires tailored integration of the people, processes, and technology in support of prevention, monitoring, and detection within an environment.

Linux malware detection and prevention is not prevalent at this time, but Palo Alto Networks customers are protected through our next-generation security platform:

IPS signature 14917 deployed to identify and prevent command and control activity

The C2 domains and files mentioned in this report are blocked in our Threat Prevention product.

Indicators

Type	Value
MD5	364ff454dcf00420cff13a57bcb78467
SHA256	8bca0031f3b691421cb15f9c6e71ce193 355d2d8cf2b190438b6962761d0c6bb
ssdeep	3072:n+1R4tREtGN4qyGCXdHPYK9l 0H786O26BmMAwyWMn/qwwiHNl:n +1R43QcILXdF0w6IBmMAwwCwwi
MD5	075b6695ab63f36af65f7ffd45cccd39
SHA-256	02c7cf55fd5c5809ce2dce56085ba437 95f2480423a4256537bfdfda0df85592
ssdeep	3072:9ZAxHANuat3WWFY9nqjwbuZf 454UNqRpROIDLHaSeWb3LGmPTrI W33HxIajF:9ZAxHANJAvbuZf454UN +rv eQLZPTrV3Z
MD5	e107c5c84ded6cd9391aede7f04d64c8
SHA-256	fd8b2ea9a2e8a67e4cb3904b49c789d 57ed9b1ce5bebfe54fe3d98214d6a0f61
ssdeep	6144:W/D5tpLWtr91gmaVy+mdckn6 BCUdc4mLc2B9:4D5Lqgkcj+
Path	/bin/rsyncd
Path Desc	synchronize and backup service
Path	~/.config/dbus-notifier/dbus-inotifier
Path Desc	system service d-bus notifier
Path	/bin/ksysdefd
Path	~/.config/ksysdef/ksysdefd
Path Desc	system kernel service defender
C2	azureon-line[.]com
C2	198.105.125[.]74
C2	mozilla-plugins[.]com
C2	Mozillaplagins[.]com

27 Pingbacks & Trackbacks

February 13, 2016 5:13 AM

The Fysbis Linux Backdoor Used By Russian Hackers - BitsHacker

February 13, 2016 1:09 PM

Fysbis Linux, El nuevo BackDoor de la Familia Malware « Tecnoinnovador

February 14, 2016 8:38 AM

Russian Hackers Spying On Your Linux PC Using Sophisticated Malware "Fysbis" | ..:: Frog in the box ::.. | ..:: Frog in the box ::..

February 15, 2016 5:56 AM

Russian cyberspy group uses simple yet effective Linux Trojan - GeekTechTalk

February 15, 2016 6:04 AM

Russian cyberspy group uses simple yet effective Linux Trojan | Network Hunt
February 15, 2016 6:05 AM
Russian cyberspy group uses simple yet effective Linux Trojan | E-Network News
February 15, 2016 7:47 AM
Russian cyberspy group uses simple yet effective Linux Trojan -
February 15, 2016 10:31 AM
| 2600 Solutions, LLC
February 15, 2016 10:32 AM
Russian cyberspy group uses simple yet effective Linux Trojan | 2600 Solutions, LLC
February 15, 2016 12:17 PM
A Look Into Fysbis: Sofacy’s Linux Backdoor | vyagers
February 15, 2016 3:17 PM
Russian Cyberespionage Group Uses Linux Trojan - LIFARS
February 15, 2016 5:08 PM
Ciberespías rusos utilizan troyano en Linux simple y efectivo | Adictec
February 15, 2016 9:10 PM
Russian cyberspy group uses simple yet effective Linux Trojan - Micro Penguin
February 15, 2016 9:27 PM
A Look Into Fysbis: Sofacy’s Linux Backdoor – Palo Alto Networks Blog | Georgia 2600 Hackers irc.2600.net #GA2600
February 16, 2016 5:10 AM
Russian cyberspy group uses simple yet effective Linux Trojan | Templar Shield
February 16, 2016 9:53 AM
Russian cyberspy group uses simple yet effective Linux Trojan | AskIT
February 16, 2016 10:40 AM
Sofacy Linux бэкдор | Around Cyber
February 17, 2016 4:56 AM
Linux Fysbis Trojan, a weapon in the Pawn Storm's arsenalSecurity Affairs
February 17, 2016 4:59 AM
Linux Fysbis Trojan, a new weapon in the Pawn Storm’s arsenal - Systerity
February 17, 2016 1:45 PM
Linux Fysbis Trojan, a new weapon in the Pawn Storm’s arsenal - OSINFO
February 17, 2016 7:06 PM
HACKERS RUSOS UTILIZAN UN SENCILLO PERO EFECTIVO TROYANO PARA ATACAR LINUX | SR HADDEN CONSULTING GROUP
February 18, 2016 9:42 PM
Russian cyberspy group uses simple yet effective Linux Trojan | The Silicon Review
February 20, 2016 2:52 PM
Weekendowa Lektura 2016-02-20 – bierzcie i czytajcie | Zaufana Trzecia Strona
February 21, 2016 6:56 PM
FYSBIS分析报告：SOFACY的Linux后门 | z7y Blog
February 24, 2016 5:01 PM
What is this Fysbis Malware? | WARLOCK
February 24, 2016 5:53 PM
What is this Fysbis Malware?- DLIT Blog
March 1, 2016 1:18 AM
FYSBIS分析报告：SOFACY的Linux后门 | 邪恶十六进制

Post Your Comment

Name *

Email *

Website

Post Comment

Home
Government
Partners
Unit 42 Threat Intelligence
Technical Documentation
Advanced Endpoint Protection

Get Updates
Sign up to receive the latest news, cyber threat intelligence and research from Unit 42.

Business Email

Submit

Subscribe to the Research Center Blog

Subscribe

Categories & Archives

Select a Category ▼ Select a Month ▼

More →

Recent Posts

Palo Alto Networks News of the Week – April 2 posted by Anna Lough on April 2, 2016
Don't Be an April Fool: Inside a Common Phone Scam posted by Robert Falcone on April 1, 2016
Channel Scoop – April 1, 2016 posted by Lang Tibbils on April 1, 2016
How the EITest Campaign's Path to Angler EK Evolved Over Time posted by Brad Duncan on March 31, 2016
Top 5 Things to Know Before You Go to Ignite Conference 2016 posted by Catherine Crandall on March 31, 2016
More →

About Palo Alto Networks

Palo Alto Networks is the network security company. Our innovative platform allows enterprises, service providers, and government entities to secure their networks and safely enable the increasingly complex and rapidly growing number of applications running on their networks.

The core of Palo Alto Networks' platform is our next-generation firewall, which delivers application, user, and content visibility and control integrated within the firewall through its proprietary hardware and software architecture. Palo Alto Networks products and services can address a broad range of network security requirements, from the datacenter to the network perimeter, as well as the distributed enterprise, which includes branch offices and a growing number of mobile devices.

FOLLOW US

Facebook
Twitter
Linked In
You Tube

Learn More

Firewalls
VPN
Malware
Intrusion Prevention System
Intrusion Detection System
Denial of Service Attack
Security Policy