

EXHIBIT 4

Security Exploit in July 18, 2017 Build

Posted Aug 7, 2017

Updated Aug 15, 2017

Kaspersky Labs has issued a press release regarding this issue along with a joint statement with NetSarang which can be read here:

https://usa.kaspersky.com/about/press-releases/2017_shadowpad-attackers-hid-backdoor-in-software-used-by-hundreds-of-large-companies-worldwide

On Friday August 4th, 2017, our engineers in cooperation with Kaspersky Labs discovered a security exploit in our software specific to the following Builds which were released on July 18, 2017. As of Aug 15, 2017, Kaspersky Labs has discovered a single instance of this exploit being utilized in Hong Kong.

Affected Builds

Xmanager Enterprise 5.0 Build 1232

Xmanager 5.0 Build 1045

Xshell 5.0 Build 1322

Xftp 5.0 Build 1218

Xlpd 5.0 Build 1220

Build numbers before and after the above Builds were not affected. If you are using any of these above listed Builds, we highly recommend you cease using the software until you update your clients. The exploit was effectively patched with the release of our latest Build on August 5th, so if you've already updated, then your clients are secure. The latest Builds are Xmanager Enterprise Build 1236, Xmanager Build 1049, Xshell Build 1326, Xftp Build 1222, and Xlpd Build 1224.

How to Update

If you are using the affected Build, you can update by going to Help -> Check for Updates directly in your client or download the latest Build from our website here: <https://www.netsarang.com/download/software.html>.

The antivirus industry has been informed of the issue and therefore your antivirus may have already quarantined/deleted the dll file which was affected. If this is the case, you will not be able to run the software. You'll need to update manually by downloading the latest build from the link posted above. Installing the updated build over your existing installation will resolve the issue.

We are working with Kaspersky Labs to further evaluate the exploit and will update our users with any pertinent information.

Products	Download	Sales	Resellers	Support	About	Contact
Xmanager Enterprise 5	Product Download	License & Service	Worldwide Resellers	Xmanager	About NetSarang	Social Networks
Xmanager 5	Font Download	Price List	Reseller Store	Xshell		Twitter
Xshell 5	Free License	Online Quote		Xftp		Facebook
Xftp 5		Online Store		Xlpd		Blog
Xlpd 5		Sales FAQ		Support Request		
		How to Buy		Customer Service		

Progress Report of the nsock2.dll Backdoor

The aftermath of "ShadowPad" and how we're moving forward

Posted Aug 30, 2017

Here at NetSarang, we're committed to the security of our users. Not only is it implicit to the nature of our software, but the industries in which our software is deployed demands it. The backdoor which was discovered in our software on August 4th, 2017, dubbed ShadowPad, was an unfortunate and costly mistake that should have never happened. Below, we'll go over what happened over the course of the last few weeks and how we've decided to move forward.

First, if you are still using the affected Build, please update your software immediately. Information on how to update your software can be found in our initial announcement here:

https://www.netsarang.com/news/security_exploit_in_july_18_2017_build.html

Discovery of the Backdoor

On August 4th, we received a tip that our software was making suspicious DNS queries. An emergency meeting for company officials was immediately called and we began an investigation. The investigation lead to the discovery of malicious code in the nsock2.dll file which was harboring a backdoor that had the potential to be exploited by its creator. Further investigation showed that it was only present in the latest Build at the time. The affected Build was taken down and an All-Hands-on-Deck meeting was called.

Phase 1 - Containment and Formal Notifications

At this point, our number 1 priority became closing the backdoor, releasing a patch, and moving users off the affected Build en masse. Acting swiftly was crucial. A new build was created and was verified clean by a third party cybersecurity firm. The build was pushed to all users, even those who were using our software illegitimately.

Users who were using the affected Build, were connected to the internet, and had opted in to receive update periodically were shown a prompt informing them of the security issue and to update their software immediately. A round of emails was sent out as well. At this point, antivirus software began quarantining/deleting the affected dll file and many users were reporting that they were unable to launch the software which could be remedied by installing the latest secure Build.

Phase 2 - Continued Investigation and Monitoring

After we released the patch, we continued to monitor the situation and received inquiries from our customers and userbase as expected. Currently, there have been no direct reports to us of any critical information being stolen. Once we ensured the security of our users' clients had been restored, our investigation turned to the source of the backdoor itself. What was clear was that we had been hit by a supply chain attack. Our network had been compromised and we were not ruling out any possibilities. We needed an independent third-party to step in and assist us in investigating the issue to ensure we were attacking this issue from all angles without bias. Therefore on August 7th, 2017, we enlisted the help of KISA (Korea Internet & Security Agency).

About four days later KISA's investigation discovered a weak link in our network security and the probable vector from which our supply chain was compromised. There are remnants of the intrusion into our network, as well as evidence of the attacker's attempts to hide their tracks. Currently, we do not know who the hacker is nor do we know what their motivation was. We are continuing to investigate and hope to one day bring the attacker to justice.

Phase 3 - Network Infrastructure Migration

After it became clear that our network infrastructure was compromised, we made the decision to abandon it completely. We cannot risk the security of our users so we began the process of migrating to a completely separate and new network infrastructure. Each device is being placed one-by-one into the new network infrastructure. The process is outlined below:

1. **Backup** - Each device is backed up and stored offline for further investigation as required.
2. **Wipe** - Once a device is backed up, it is wiped completely and prepped for introduction into our new network infrastructure.
3. **Examine** - Before the device can be put into our new infrastructure, it has all the necessary components installed. The device is examined by multiple individuals ensuring that there are no lingering issues.
4. **Approval** - The device goes through a final approval process and is signed off to be placed into our new network infrastructure.

Through this method, we can be sure that we're re-starting our development with a clean slate. As of August 28th, 2017, we've completed roughly half of the migration.

Phase 4 - Back to Work

Since critical machines were cleared for introduction into the new network infrastructure first, we were able to slowly get back to development. Our Code Signing certificate has been reissued, and on August 25th 2017, we released another Build of our full line of software. We upgraded session file password encryption and have separated our license types into separate packages which improves deployment efficiency of our software.

We're also improving our development policies to ensure we never again deliver a compromised package to our users. Before each release, there are a number of checks we use to make sure the outputted package from our Build machine is secure and contains nothing we don't want included in it. Our policies include multiple checks and comparisons of source code by multiple individuals, additional antivirus scans across multiple antivirus tools via VirusTotal, a set duration pre-run monitoring of new Builds before they are released to the public, and other internal procedures.

What We Learned

The past few weeks have been rough. Not just for us, but even more so for our users. Many users have contacted us to let us know that they are moving away from our software, and we are saddened to see these users go. However, others have reached out to us with messages of encouragement and have expressed their faith in our ability to overcome and learn from our mistakes for which we are unimaginably grateful. We will work tirelessly to restore our reputation.

Thinking that one is 100% secure is hubris. Cyberattacks are continuously evolving, and thus we need to be ever-changing as well. Our hope is that ShadowPad can be source of education for organizations around the world. It is our responsibility as software developers to anticipate and preemptively prepare for what's coming next.

As a user of NetSarang software, you have our promise that we are monitoring the situation with the utmost priority. We will utilize it as a learning experience from which we must improve and regain your trust. Thank you for using NetSarang software.

Products	Download	Sales	Resellers	Support	About	Contact
Xmanager Enterprise 5	Product Download	License & Service	Worldwide Resellers	Xmanager	About NetSarang	Social Networks
Xmanager 5	Font Download	Price List	Reseller Store	Xshell		Twitter
Xshell 5	Free License	Online Quote		Xftp		Facebook
Xftp 5		Online Store		Xlpd		Blog
Xlpd 5		Sales FAQ		Support Request		
		How to Buy		Customer Service		


10/20/2017

ShadowPad in corporate networks - Securelist

corporate networks

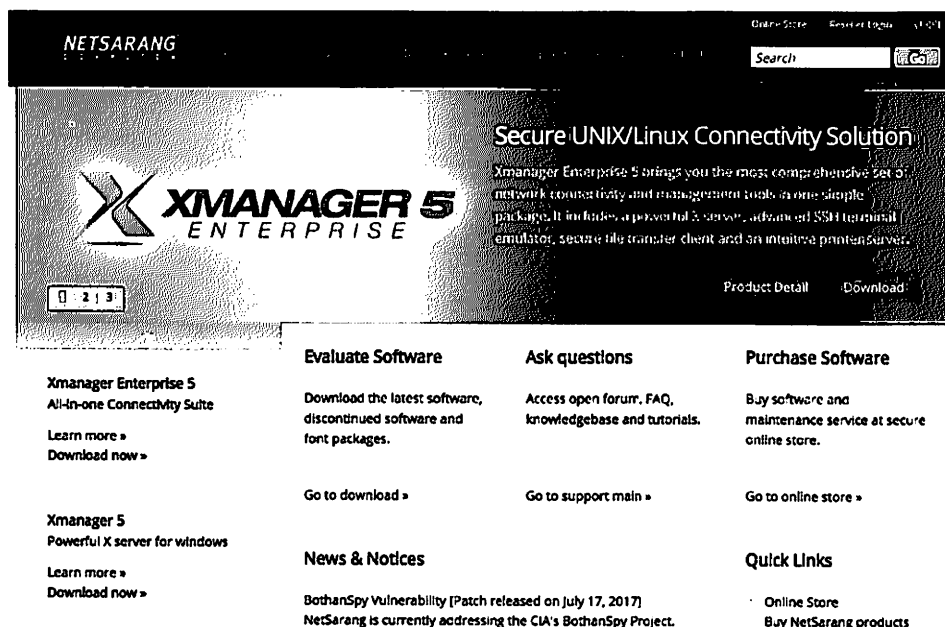
Popular server management software hit in supply chain attack

By GReAT on August 15, 2017. 6:00 pm

 ShadowPad, part 2: Technical Details (PDF)

In July 2017, during an investigation, suspicious DNS requests were identified in a partner's network. The partner, which is a financial institution, discovered the requests originating on systems involved in the processing of financial transactions.

Further investigation showed that the source of the suspicious DNS queries was a software package produced by NetSarang. Founded in 1997, NetSarang Computer, Inc. develops, markets and supports secure connectivity solutions and specializes in the development of server management tools for large corporate networks. The company maintains headquarters in the United States and South Korea.



The screenshot shows the NetSarang website interface. At the top, there is a navigation bar with the NetSarang logo, a search bar, and links for 'Create Store', 'Event Login', and 'VMS'. Below the navigation bar is a large banner for 'Xmanager 5 Enterprise' with the text 'Secure UNIX/Linux Connectivity Solution'. The banner includes a description: 'Xmanager Enterprise 5 brings you the most comprehensive set of network connectivity and management tools in one simple package. It includes a powerful X server, advanced SSH terminal emulator, secure file transfer client and an intuitive printer server.' Below the banner are four main sections: 'Evaluate Software' (Download the latest software, discontinued software and font packages. Go to download >), 'Ask questions' (Access open forum, FAQ, knowledgebase and tutorials. Go to support main >), 'Purchase Software' (Buy software and maintenance service at secure online store. Go to online store >), and 'News & Notices' (BothanSpy Vulnerability [Patch released on July 17, 2017] NetSarang is currently addressing the CIA's BothanSpy Project.). There is also a 'Quick Links' section with 'Online Store' and 'Buy NetSarang products'.

NetSarang website

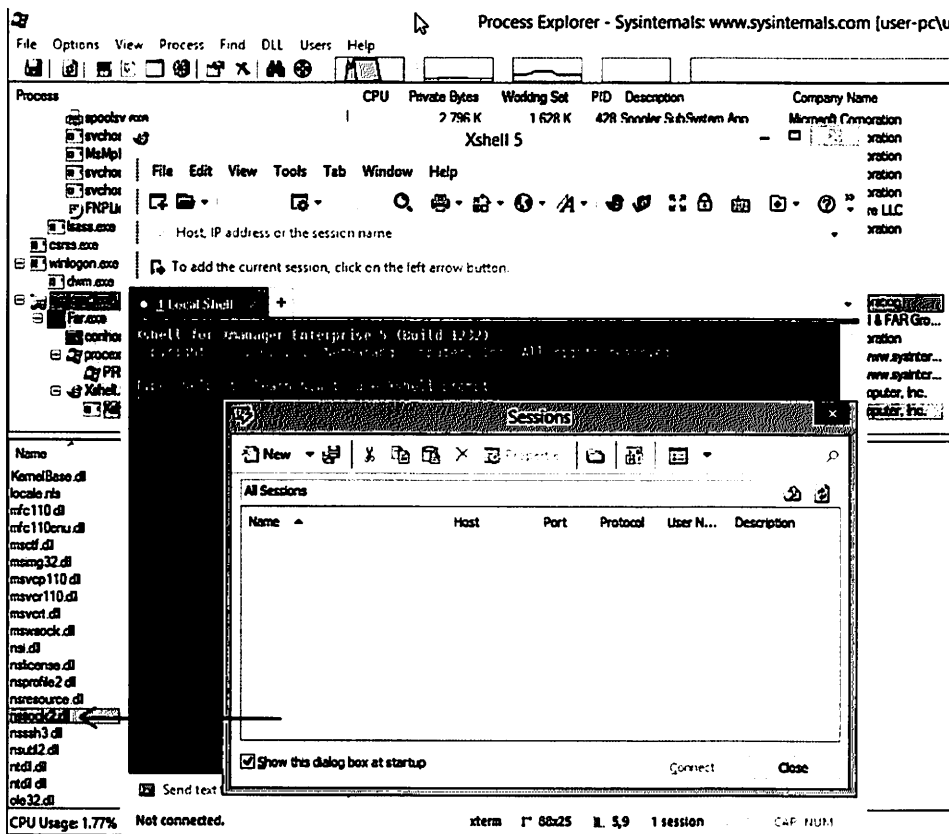
Our analysis showed that recent versions of software produced and distributed by NetSarang had been surreptitiously modified to include an

10/20/2017

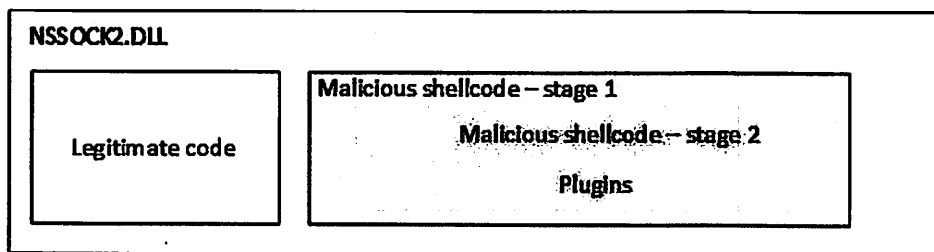
ShadowPad in corporate networks - Securelist

encrypted payload that could be remotely activated by a knowledgeable attacker.

The backdoor was embedded into one of the code libraries used by the software (nsock2.dll):



Backdoored dll in a list of loaded modules of Xshell5 software



Disposition of the NSSOCK2.DLL binary with embedded malicious code

The attackers hid their malicious intent in several layers of encrypted code. The tiered architecture prevents the actual business logics of the backdoor from being activated until a special packet is received from the first tier command and control (C&C) server ("activation C&C server"). Until then, it only transfers basic information, including the computer, domain and user names, every 8 hours.

10/20/2017

ShadowPad in corporate networks - Securelist

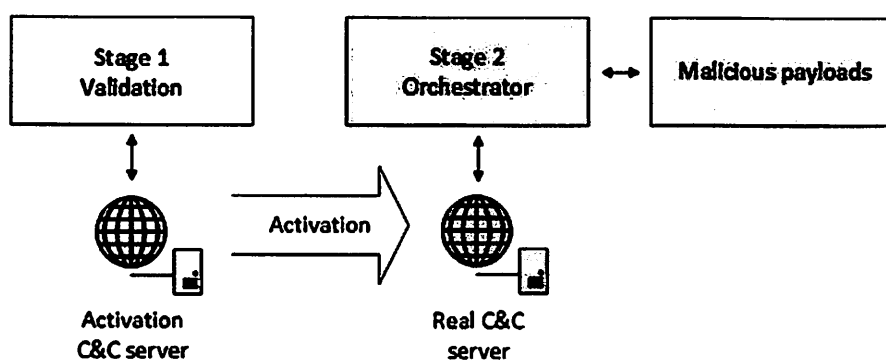
Activation of the payload would be triggered via a specially crafted DNS TXT record for a specific domain. The domain name is generated based on the current month and year values, e.g. for August 2017 the domain name used would be "nylalobghyhirgh.com".

```

Internet Protocol Version 4, Src: [REDACTED], Dst: 8.8.8.8
User Datagram Protocol, Src Port: 50242 (50242), Dst Port: 53 (53)
Domain Name System (query)
Transaction ID: 0x6ff2
Flags: 0x0100 Standard query
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
Queries
  [REDACTED] qoolykc.jkrdrgwckpq.nylalobghyhirgh.com: type TXT, class IN
    Name: [REDACTED] qoolykc.jkrdrgwckpq.nylalobghyhirgh.com
    [Name Length: 84]
    [Label Count: 4]
    Type: TXT (Text strings) (16)
    Class: IN (0x0001)

```

DNS queries to C&C from backdoored nsock2.dll



Only when triggered by the first layer of C&C servers does the backdoor activate its second stage

The module performs a quick exchange with the controlling DNS server and provides basic target information (domain and user name, system date, network configuration) to the server. The C&C DNS server in return sends back the decryption key for the next stage of the code, effectively activating the backdoor. The data exchanged between the module and the C&C is encrypted with a proprietary algorithm and then encoded as readable latin characters. Each packet also contains an encrypted "magic" DWORD value "52 4F 4F 44" ('DOOR' if read as a little-endian value).

Our analysis indicates the embedded code acts as a modular backdoor platform. It can download and execute arbitrary code provided from the C&C server, as well as maintain a virtual file system (VFS) inside the registry. The VFS, and any additional files created by the code, are encrypted and stored in a location unique to each victim. The remote access capability includes a domain generation algorithm (DGA) for C&C servers which changes every month. The attackers behind this malware have already

10/20/2017

ShadowPad in corporate networks - Securelist

registered the domains covering July to December 2017, which indirectly confirms alleged start date of the attack as around mid July 2017.

Currently, we can confirm activated payload in a company in Hong Kong. Given that the NetSarang programs are used in hundreds of critical networks around the world, on servers and workstations belonging to system administrators, it is **strongly** recommended that companies take immediate action to identify and contain the compromised software.

Kaspersky Lab products detect and protect against the backdoored files as "Backdoor.Win32.ShadowPad.a".

We informed NetSarang of the compromise and they immediately responded by pulling down the compromised software suite and replacing it with a previous clean version. The company has also published a message acknowledging our findings and warning their customers.

ShadowPad is an example of the dangers posed by a successful supply-chain attack. Given the opportunities for covert data collection, attackers are likely to pursue this type of attack again and again with other widely used software components. Luckily, NetSarang was fast to react to our notification and released a clean software update, most likely preventing hundreds of data-stealing attacks against their clients. This case is an example of the value of threat research as a means to secure the wider internet ecosystem. No single entity is in a position to defend all of the links in an institution's software and hardware supply-chain. With successful and open cooperation, we can help weed out the attackers in our midst and protect the internet for all users, not just our own.

For more information please contact: intelreports@kaspersky.com

Frequently Asked Questions

What does the code do if activated?

If the backdoor were activated, the attacker would be able to upload files, create processes, and store information in a VFS contained within the victim's registry. The VFS and any additional files created by the code are encrypted and stored in locations unique to each victim.

Which software packages were affected?

10/20/2017

ShadowPad in corporate networks - Securelist

We have confirmed the presence of the malicious file (nsock2.dll) in the following packages previously available on the NetSarang site:

Xmanager Enterprise 5 Build 1232

Xme5.exe, Jul 17 2017, 55.08 MB

MD5: 0009f4b9972660eeb23ff3a9dccc8d86

SHA1: 12180ff028c1c38d99e8375dd6d01f47f6711b97

Xmanager 5 Build 1045

Xmgr5.exe, Jul 17 2017, 46.2 MB

MD5: b69ab19614ef15aa75baf26c869c9cdd

SHA1: 35c9dae68c129ebb7e7f65511b3a804ddbe4cf1d

Xshell 5 Build 1322

Xshell5.exe, Jul 17 2017, 31.58 MB

MD5: b2c302537ce8fbbcff0d45968cc0a826

SHA1: 7cf07efe04fe0012ed8beaa2dec5420a9b5561d6

Xftp 5 Build 1218

Xftp5.exe, Jul 17 2017, 30.7 MB

MD5: 78321ad1deefce193c8172ec982ddad1

SHA1: 08a67be4a4c5629ac3d12f0fdd1efc20aa4bdb2b

Xlpd 5 Build 1220

Xlpd5.exe, Jul 17 2017, 30.22 MB

MD5: 28228f337fdbe3ab34316a7132123c49

SHA1: 3d69fdd4e29ad65799be33ae812fe278b2b2dabe

Is NetSarang aware of this situation?

Yes, we contacted the vendor and received a swift response. Shortly after notification by Kaspersky Lab all malicious files were removed from NetSarang website.

How did you find the software was backdoored?

During an investigation, suspicious DNS requests were identified on a partner's network. The partner, which is a financial institution, detected these requests on systems related to the processing of financial transactions. Our analysis showed that the source of these suspicious requests was a software package produced by NetSarang.

When did the malicious code first appear in the software?

A fragment of code was added in nsock2.dll (MD5: 97363d50a279492fda14cbab53429e75), compiled Thu Jul 13 01:23:01 2017. The file is signed with a legitimate NetSarang certificate (Serial number: 53 0C E1 4C 81 F3 62 10 A1 68 2A FF 17 9E 25 80). This code is not present in the nsock2.dll from March (MD5: ef0af7231360967c08efbdd2a94f9808) included with the NetSarang installation kits from April.

How do I detect if code is present on a system?

All Kaspersky Labs products detect and cure this threat as Backdoor.Win32.Shadowpad.a. If for some reason you can't use an antimalware solution you can check if there were DNS requests from your organization to these domains:

- ribotqtonut[.]com
- nyalobghyhirgh[.]com
- jkvmdmjyfcvkf[.]com
- bafyvoruzgjitwr[.]com
- xmponmzmxkxkh[.]com
- tczafklirkl[.]com
- notped[.]com
- dnsgogle[.]com
- operatingbox[.]com
- paniesx[.]com
- technicianetext[.]com

How do I clean any affected systems?

All Kaspersky Lab products successfully detect and disinfect the affected files as "Backdoor.Win32.Shadowpad.a" and actively protect against the threat.

If you do not have a Kaspersky product installed, then:

1. Update to the latest version of the NetSarang package.
2. Block DNS queries to the C2 domains listed in Appendix A.

What kind of companies/organizations/are targeted by the attackers?

10/20/2017

ShadowPad in corporate networks - Securelist

Based on the vendor profile, the attackers could be after a broad set of companies who rely on NetSarang software, which includes banking and financial industry, software and media, energy and utilities, computers and electronics, insurance, industrial and construction, manufacturing, pharmaceuticals, retail, telecommunications, transportation and logistics and other industries.

Who is behind this attack?

Attribution is hard and the attackers were very careful to not leave obvious traces. However certain techniques were known to be used in another malware like PlugX and Winnti, which were allegedly developed by Chinese-speaking actors.

How did the attackers manage to get access to create trojanized updates. Does that mean that NetSarang was hacked?

An investigation is in progress, but since code was signed and added to all software packages it could point to the fact that attackers either modified source codes or patched software on the build servers.

Appendix A – Indicators of Compromise

At this time, we have confirmed the presence of the malicious "nsock2.dll" in the following packages downloaded from the NetSarang site:

Xmanager Enterprise 5 Build 1232
Xme5.exe, Jul 17 2017, 55.08 MB
MD5: 0009f4b9972660eeb23ff3a9dccd8d86
SHA1: 12180ff028c1c38d99e8375dd6d01f47f6711b97

Xmanager 5 Build 1045
Xmgr5.exe, Jul 17 2017, 46.2 MB
MD5: b69ab19614ef15aa75baf26c869c9cdd
SHA1: 35c9dae68c129ebb7e7f65511b3a804ddbe4cf1d

Xshell 5 Build 1322
Xshell5.exe, Jul 17 2017, 31.58 MB

10/20/2017

ShadowPad in corporate networks - Securelist

MD5: b2c302537ce8fbbcff0d45968cc0a826
SHA1: 7cf07efe04fe0012ed8beaa2dec5420a9b5561d6

Xftp 5 Build 1218
Xftp5.exe, Jul 17 2017, 30.7 MB
MD5: 78321ad1deefce193c8172ec982ddad1
SHA1: 08a67be4a4c5629ac3d12f0fdd1efc20aa4bdb2b

Xlpd 5 Build 1220
Xlpd5.exe, Jul 17 2017, 30.22 MB
MD5: 28228f337fdba3ab34316a7132123c49
SHA1: 3d69fdd4e29ad65799be33ae812fe278b2b2dabe

Domains:

ribotqtonut[.]com
nylalobghyhirgh[.]com
jkvmdmjyfcvkf[.]com
bafyvoruzgjitwr[.]com
xmponmzmxkxkh[.]com
tczafklirk[.]com
notped[.]com
dnsgogle[.]com
operatingbox[.]com
paniesx[.]com
techniciantext[.]com

DLL with the encrypted payload:

97363d50a279492fda14cbab53429e75

NetSarang packages which contain the DLL with the encrypted payload (same as above, just the list of MD5 sums):

0009f4b9972660eeb23ff3a9dccd8d86
b69ab19614ef15aa75baf26c869c9cdd
b2c302537ce8fbbcff0d45968cc0a826
78321ad1deefce193c8172ec982ddad1
28228f337fdba3ab34316a7132123c49

File names:

10/20/2017

ShadowPad in corporate networks - Securelist

nsock2.dll

BACKDOOR DNS SOFTWARE SUPPLY-CHAIN ATTACK

Share post on:

TARGETED ATTACKS

Related Posts

ATMii: a small but effective ATM robber

The Festive Complexities of SIGINT-Capable Threat Actors

A Modern Hypervisor as a Basis for a Sandbox

THERE ARE 3 COMMENTS

Ken Hollis

Posted on August 16, 2017. 8:11 pm

Since the domains have apparently disappeared can you give the IP address that they were last seen at?

Thanks

REPLY

Nima

Posted on August 16, 2017. 11:22 pm

this is only security exploit , not backdoor .

REPLY

Patrick Hunter

Posted on September 8, 2017. 8:36 pm

There's malicious code actually inserted into the software which is then activated later; that is absolutely a backdoor. This isn't just some RCE exploit.

REPLY



ShadowPad: popular server management software hit in supply chain attack

Part 2: Technical Details



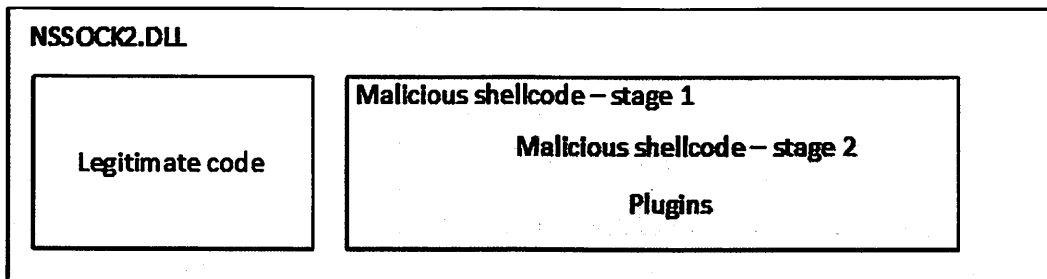
Kaspersky Lab

ShadowPad is a modular cyber-attack platform that attackers deploy in victim networks to gain flexible remote control capabilities. The platform is designed to run in two stages. The first stage is a shellcode that was embedded in a legitimate nsock2.dll used by Xshell, Xmanager and other software packages produced by NetSarang. This stage is responsible for connecting to "validation" command and control (C&C) servers and getting configuration information including the location of the real C&C server, which may be unique per victim. The second stage acts as an orchestrator for five main modules responsible for C&C communication, working with the DNS protocol, loading and injecting additional plugins into the memory of other processes.

All actual payloads are received from the real C&C as plugins and can perform different types of data exfiltration.

NSSOCK2.DLL - the compromised library

```
SHA256      462a02a8094e833fd456baf0a6d4e18bb7dab1a9f74d5f163a8334921a4ffde8
MD5         97363d50a279492fda14cbab53429e75
Compiled    2017.07.13 01:23:01 (GMT), 11.0
Type        I386 Windows GUI DLL
Size        180432
Internal name nsock2.dll
```



The main loader is built into the original "nsock2.dll", which is digitally signed. The malicious code is triggered from one of the object autoinitializations that are automatically called by the C runtime code. It decrypts a binary blob with a function similar to "rand" and directly starts its execution.

The blob is a self-loading executable converted into shellcode. It starts with a loader that processes a proprietary PE-like formatted blob, loads the code and data section by section, resolves imported API functions, relocates the code and then calls the entrypoint as DllMain.

Each self-loading shellcode contains a timestamp field that appears to be equal to UNIX timestamps.

Shellcode in NSSOCK2.DLL

```
Size        77824
Type        shellcode, binary reconstructed from a proprietary format
```

Kaspersky Lab

Timestamp 2017.05.26 07:00:23 (GMT)

The binary object is produced from a compiled Windows DLL file. The entrypoint of the blob starts with a standard Microsoft Visual Studio DllMain code stub. All strings are encrypted with a custom randomisation function and each string starts with a 2-byte decryption key. It maintains a configuration block in the Windows registry using one of the following locations:

HKCU\SOFTWARE\%d

or

HKLM\SOFTWARE\%d,

where *%d* is a signed integer produced from the system drive's serial number *xor-ed* with 0xD592FC92. The block is stored in a value named "Data" and is 552 bytes long. It contains a unique user id (generated GUID), 8 byte decryption key for the second stage, first execution time, execution counter. It generates a hostname for accessing its C&C server using a DGA (domain generation algorithm) based on the current month and year in the .com top level domain. The request to the C&C is sent through the DNS extracted from the network adapter settings or to hardcoded DNS servers IPs : 8.8.8.8, 8.8.4.4, 4.2.2.1, 4.2.2.2.

As of August 2017, the following domain name was used: **nylalobghyhirgh.com**

At the time of analysis the domain was registered with the following WHOIS information:

```
Domain Name: NYLALOBGHYHIRGH.COM
Registry Domain ID: 2146218329_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.namesilo.com
Registrar URL: http://www.namesilo.com
Updated Date: 2017-07-24T06:41:22Z
Creation Date: 2017-07-24T06:41:22Z
Registry Expiry Date: 2018-07-24T06:41:22Z
Registrar: NameSilo, LLC
Registrar IANA ID: 1479
Registrar Abuse Contact Email: abuse@namesilo.com
Registrar Abuse Contact Phone: +1.4805240066
Domain Status: clientTransferProhibited
https://icann.org/epp#clientTransferProhibited
Name Server: NS1.QHOSTER.NET
Name Server: NS2.QHOSTER.NET
Name Server: NS3.QHOSTER.NET
Name Server: NS4.QHOSTER.NET
DNSSEC: unsigned
```

DNS requests are sent every 8 hours. The request buffer presented to the C&C server contains the following data:

Value	Description
00 00	encryption key

Kaspersky Lab

```

52 4F 4F 44      'DOOR', magic value
6 bytes         GUID
2 bytes         iteration counter
1 byte          year's least significant byte + 0x30
1 byte          month
1 byte          day
*               hostname
*               domain name
*               user name

```

The request buffer is encrypted with custom XOR-based encryption algorithm. Then, the encrypted buffer is converted to a readable string of latin characters by adding each half of the byte to 'a' and 'j' characters correspondingly.

The first character is encoded by adding the 'a' character to the number of non-dot characters in the DGA domain name. This string is split in a series of subdomains of 50-63 bytes long split by dots and then prepended to the DGA-generated hostname name.

The resulting request packet querying a `*.....*.%DGA-domain%.com` is sent to all the DNS servers available and Google DNS servers.

The DNS packet wrapping the request buffer starts with the following fields:

Value	Description
2 bytes	Random request ID
01 00	Opcode (recursive request)
00 01	Number of queries: 1
00 00	Number of answers: 0
00 00	Number of name server records: 0
00 00	Number of authoritative response records: 0
*	Encoded data represented as a hostname
00 10	Query type: TXT
00 01	Query class: IN

The module waits for a response from any DNS server until timed out or some data is received. The DNS packet is checked to conform to the following format:

Value	Description
?? ??	ID
xx x0	No error
xx xx	Number of queries
xx xx	Number of answers
xx xx	Number of name server records
00 00	Number of authoritative response records

Kaspersky Lab

```

--query--
*           Encoded data in the query record, first char + 'a'
is number of stray bytes at the end, all the rest encoded as two
latin characters starting from 'a', 'j'
00 10      Query type: TXT
00 01      Query class: IN
--response--
C0 0C      Name: backwards link to the query record
00 10      Query type: TXT
00 01      Query class: IN
00 01 xx xx  TTL
xx xx      Record length
*           Encoded response from the C2 server. Data format is
the same as for the query part.

```

The response string is decrypted using the first two bytes of the response packet as a key.
The data format follows:

```

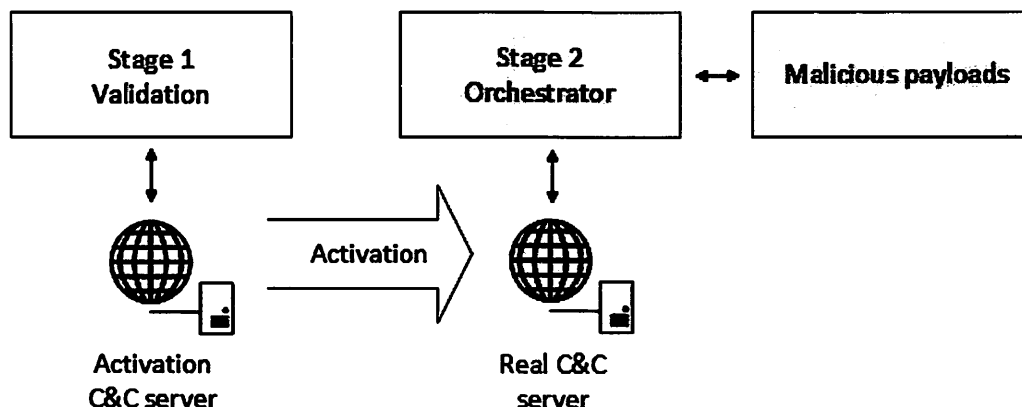
xx xx      encryption key
-- after decryption --
52 4F 4F 44  'DOOR', magic value
2 bytes     iteration counter
1 byte     status : 1 - ready to decrypt the payload, 2 - stop
operation
4 bytes     part of decryption key
4 bytes     part of decryption key
4 bytes     length of additional data
*           additional data

```

The module copies information received from the C&C server to its configuration storage and updates the corresponding registry key.

Once a proper decryption key is read from the registry or from the DNS response it is used to decipher the second encrypted shellcode ("stage 2"). It is then called directly passing the 'additional data' string received from the C&C as an argument to the shellcode.

Kaspersky Lab



Second stage shellcode in “NSSOCK2.DLL”

Size	54288
Type	shellcode, binary reconstructed from a proprietary format
Timestamp	2017.05.26 07:00:13 (GMT)

The shellcode was produced from a Windows DLL file. The entrypoint of the blob starts with standard MSVC DllMain code. The format of the blob is the same as in the "Shellcode in NSSOCK2.DLL".

The DllMain function differs from the standard C/C++ implementation. Besides standard "fdwReason" parameter values it also processes custom ones: 100, 101, 102, 103, 104.

"Reason" codes 102-104 are used to implement a custom plugin API.

Code 100 : plugin initialization

Code 101 : plugin deinitialization

Code 102 : return the plugin's numeric identifier 100 in the IpReserved parameter

Code 103 : allocate a string for the plugin's name i.e. "Root" and return the value in the IpReserved parameter

Code 104 : return a pointer to plugin's function table in a DWORD pointed by the "IpReserved" parameter

During DLL initialization the module allocates memory for internal structures and sets up its function table. Then, it uses the pointer to its own image as a plugin and initializes the plugin infrastructure.

The plugin is started by calling sequentially its entrypoint (DllMain) with "Reason" parameters 100, 102, 104 and copying the data returned in a structure describing the plugin. Once the plugin returns no error during initialization it is added to the plugin list.

Depending on the mode of operation, it can then proceed with the plugin orchestrator, either in a separate thread or inline - that is specified by the parameter provided by the C&C.

Kaspersky Lab

Module / plugin orchestration

The image contains five encrypted blobs structured in the same way as the second stage blob. They are decrypted with a XOR-based algorithm, decompressed with QuickLZ and loaded in memory. Then they are initialized and added to the list of plugins in the same way as the "Root" plugin.

Each plugin has a name and a numeric identifier (ID):

ID	Name
100	Root (the second stage shellcode itself)
101	Plugins
102	Config
103	Install
104	Online
203	DNS

Then the module searches for the plugin with ID 103 ("Install") and calls its second function. The process is terminated if the plugin is not available. The module remains in memory as the "Root" plugin and provides various facilities for the other plugins via the exported function table.

Modules

"Install" module

Size	7877
Type	shellcode, binary reconstructed from a proprietary format
Timestamp	2017.05.26 06:59:46 (GMT)

Starts by adjusting process privileges, then invokes the "Config" plugin's function "LoadConfig"

If there are no additional parameter from the C&C it continues to the main thread, otherwise it injects into a newly created process and continues from there.

Creates mutex "**Global%16-48 random latin characters%**"

The module continues by invoking the modules "Plugins" and "Online".

"Plugins" module

Size	7119
Type	shellcode, binary reconstructed from a proprietary format
Timestamp	2017.05.26 06:59:07 (GMT)

Kaspersky Lab

This plugin provides API for other modules and does not initiate any actions without external intervention.

Starts a registry monitoring thread waiting for changes in its registry key and loads any new plugins available in the registry Virtual File System (VFS).

The registry location is :

HKLM\HKCU\SOFTWARE\Microsoft\%5-12 random characters%

Every value found in the key is decrypted and checked if it is a valid plugin and then loaded and initialized using the API from the "Root" module.

Also, the plugin provides an API for reading, writing and deleting arbitrary registry values. This also allows for the writing of new plugin images to the registry VFS by command from the C&C server.

"Config" module

Size	6574
Type	shellcode, binary reconstructed from a proprietary format
Timestamp	2017.05.26 06:59:16 (GMT)

This module maintains a configuration block of data of a fixed size of 2136 bytes. The block consists of a fixed size header and a string pool populated sequentially and referenced from the fixed header. When invoked for the first time during the current session it initializes with a default configuration. The default C&C server URL may be overwritten with the one provided from the the packet used to activate the second stage shellcode, if present.

The configuration string pool starts from offset 0x58 and holds several string parameters. Each string is encrypted with a random 2-byte key using a proprietary algorithm based on XOR and an in-house rand() function. The encryption algorithm is the same for all string constants used in all of the components of the malware.

Offset	Size	Value
000	2	Offset of the string constant "HD"
002	2	Offset of the string constant "HD"
010	2	Offset of the executable path used for injection
018	2	Offset of the C&C server URL
040	4	IP address "8.8.8.8"
044	4	IP address "8.8.4.4"
048	4	IP address "4.2.2.1"
04C	4	IP address "4.2.2.2"
050	4	Constant 0x708 - sleep interval, equal to 1800 seconds or 30 minutes
058	*	String pool:

Kaspersky Lab

"HD" String constant
 "HD" String constant
 * C&C server URL, default value: "dns://www.notped.com", or the one provided by the server to the "Shellcode in NSSOCK2.DLL" code.
 "%windir%\system32\svchost.exe" Path to the executable used as a host for injection

The configuration block is prepended with a service header, compressed and encrypted using a function provided by the "Root" plugin and then written to a file. The resulting size of the file is 2156 bytes.

Service header format:

Offset	Size	Value
000	4	00 00 00 00
004	4	12 34 56 78
008	4	00 00 00 00
00C	4	00 00 08 58 // Size of the configuration block in bytes
010	4	* // Unused

The exact location of the configuration file depends on the system volume's serial number and is generated according to the following format:

%ALLUSERSPROFILE%\%random 3-8 latin characters%\%random 3-8 latin characters%\%random 3-8 latin characters%\%random 3-8 latin characters%

The file is always overwritten every time the plugin is initialized.

"Online" module

Size	15803
Type	shellcode, binary reconstructed from a proprietary format
Timestamp	2017.05.26 06:59:21 (GMT)

Handles overall communication with the C&C server and dispatches the commands to other plugins.

Maintains a registry key : ***HKLM\HKCU\SOFTWARE\%random 3-8 latin characters%*** containing a 24-byte record of system time and number of tries.

Processes the list of C&C URLs from the configuration block (up to 16 URLs). Depending on the protocol specified in the URL it selects one of the plugins for handling communication with the C&C server.

Protocol	Plugin ID

Kaspersky Lab

TCP	200
HTTP	201
HTTPS	204
UDP	202
DNS	203
SSL	205
URL	built in, DGA-based HTTP client

It maintains a connection using one of the plugins, starting the connection with an initial packet and getting and executing commands and sending result packets back. The data received back is either a shutdown message or a packet of data containing a plugin ID and additional data for the command.

In case of the "URL" protocol, it uses a built-in HTTP client to resolve the actual C&C server URL from an intermediary C&C server. It uses its own DGA based on the day of the month, range (1-10, 11-20, >20) to generate the name of the intermediary C&C server. The actual name of the server is based on a mask specified in the configuration data, i.e. *prefix%DGA-generated part%suffix*, and the location of the suffix is marked with a '@' char.

Depending on the URL scheme specified in the mask it selects FTP, HTTP or HTTPS protocol to send the request to the intermediary server and either sends a "GET" request or fetches a file from FTP.

Once it has received a response the module looks for a string framed with '\$' characters. The string is then decoded into a binary buffer by subtracting 'a' characters and concatenating each pair into one byte. Then the buffer is decrypted using an algorithm that is used for string encryption in the rest of the code. The resulting decrypted buffer is expected to be the actual URL of the C&C to use then.

The module may also provide basic information about the system when requested by the C&C server:

- current date and time
- memory status
- CPU frequency
- amount of free disk space
- video mode
- system locale
- PID of the malicious process
- OS version
- domain name
- user name

"DNS" module

Size	10982
Type	shellcode, binary reconstructed from a proprietary format
Timestamp	2017.05.26 06:58:11 (GMT)

Kaspersky Lab

Handles all C&C communication based on the DNS protocol.

Sends and receives DNS TXT records messages in the same way as the "Shellcode in NSSOCK2". However, the encoded payload is decrypted using a different in-house algorithm and the format of the response buffer is different:

Offset	Size	Value
000	2	Encryption key -- after decryption --
002	2	Packet type (0,1,3)
004	2	packet id1 (of the server's response)
006	2	packet id2 (of the packet server is responding to, ACK)

Initial packet, type 0:

Offset	Size	Value
000	2	Encryption key
002	2	00 00
004	2	packet id 1
006	2	packet id 2
008	16	GUID

Packet type 1 - data:

Offset	Size	Value
000	2	Encryption key
002	2	00 01
004	2	packet id 1
006	2	packet id 2
008	*	payload

Packet type 3 - shutdown message:

Offset	Size	Value
000	2	Encryption key
002	2	00 03
004	2	packet id 1
006	2	packet id 2