Jun 15 2023

NETMANAGEIT

Intelligence Report

CVE-2017-9248 Exploitation in U.S. Government IIS Server

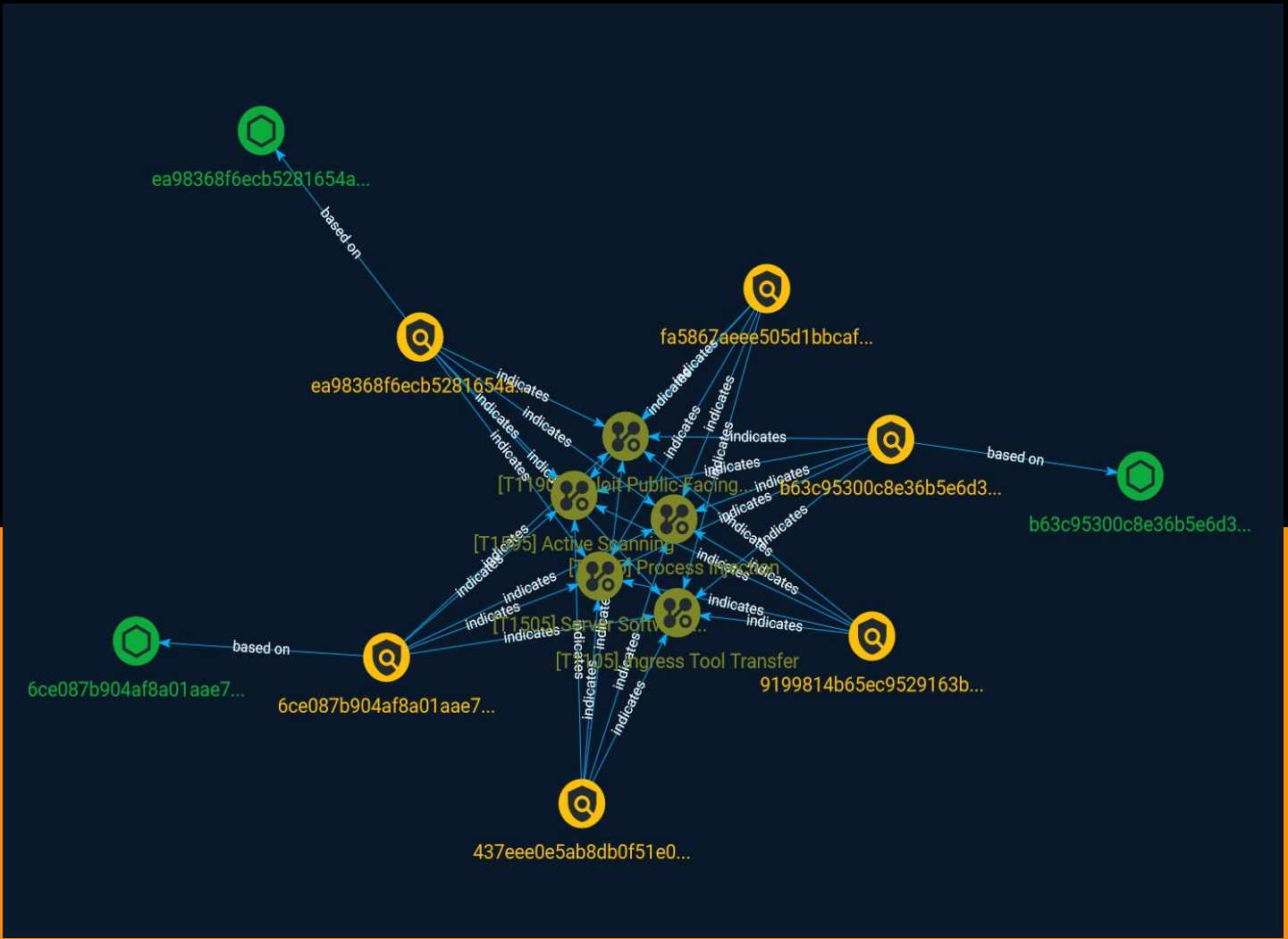

Table of contents

Overview

Entities

Observables

External References

Overview

Description

CISA received three files for analysis. The files included three webshells written in PHP: Hypertext Preprocessor (PHP), Active Server Pages Extended (ASPX), and .NET Dynamic-Link Library (DLL). The sample “sd.php” is highly obfuscated and uses rot13 algorithm, zlib for compression and base64 encoding for obfuscation. The “osker.aspx” webshell code was padded with junk code. The .NET DLL webshell is a .NET compiled version of osker.aspx. The samples are interactive webshells and have the ability to upload and manage files, create directories and files, and execute commands on the target machine.

Confidence

This value represents the confidence in the correctness of the data contained within this report.

15 / 100

Attack-Pattern

Name
Process Injection
ID
T1055
Description
Adversaries may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges. Process injection is a method of executing arbitrary code in the address space of a separate live process. Running code in the context of another process may allow access to the process's memory, system/network resources, and possibly elevated privileges. Execution via process injection may also evade detection from security products since the execution is masked under a legitimate process. There are many different ways to inject code into a process, many of which abuse legitimate functionalities. These implementations exist for every major OS but are typically platform specific. More sophisticated samples may perform multiple process injections to segment modules and further evade detection, utilizing named pipes or other inter-process communication (IPC) mechanisms as a communication channel.
Name
Server Software Component
ID
T1505

Description

Adversaries may abuse legitimate extensible development features of servers to establish persistent access to systems. Enterprise server applications may include features that allow developers to write and install software or scripts to extend the functionality of the main application. Adversaries may install malicious components to extend and abuse server applications.(Citation: volexity_0day_sophos_FW)

Name

Exploit Public-Facing Application

ID

T1190

Description

Adversaries may attempt to exploit a weakness in an Internet-facing host or system to initially access a network. The weakness in the system can be a software bug, a temporary glitch, or a misconfiguration. Exploited applications are often websites/web servers, but can also include databases (like SQL), standard services (like SMB or SSH), network device administration and management protocols (like SNMP and Smart Install), and any other system with Internet accessible open sockets.(Citation: NVD CVE-2016-6662)(Citation: CIS Multiple SMB Vulnerabilities)(Citation: US-CERT TA18-106A Network Infrastructure Devices 2018)(Citation: Cisco Blog Legacy Device Attacks)(Citation: NVD CVE-2014-7169) Depending on the flaw being exploited this may also involve [Exploitation for Defense Evasion](https://attack.mitre.org/techniques/T1211). If an application is hosted on cloud-based infrastructure and/or is containerized, then exploiting it may lead to compromise of the underlying instance or container. This can allow an adversary a path to access the cloud or container APIs, exploit container host access via [Escape to Host](https://attack.mitre.org/techniques/T1611), or take advantage of weak identity and access management policies. Adversaries may also exploit edge network infrastructure and related appliances, specifically targeting devices that do not support robust host-based defenses.(Citation: Mandiant Fortinet Zero Day)(Citation: Wired Russia Cyberwar) For websites and databases, the OWASP top 10 and CWE top 25 highlight the most common web-based vulnerabilities. (Citation: OWASP Top 10)(Citation: CWE top 25)

Name

Ingress Tool Transfer

ID

T1105

Description

Adversaries may transfer tools or other files from an external system into a compromised environment. Tools or files may be copied from an external adversary-controlled system to the victim network through the command and control channel or through alternate protocols such as [ftp](https://attack.mitre.org/software/S0095). Once present, adversaries may also transfer/spread tools between victim devices within a compromised environment (i.e. [Lateral Tool Transfer](https://attack.mitre.org/techniques/T1570)). Files can also be transferred using various [Web Service](https://attack.mitre.org/techniques/T1102)s as well as native or otherwise present tools on the victim system.(Citation: PTSecurity Cobalt Dec 2016) On Windows, adversaries may use various utilities to download tools, such as `copy`, `finger`, [certutil](https://attack.mitre.org/software/S0160), and [PowerShell](https://attack.mitre.org/techniques/T1059/001) commands such as `IEX(New-Object Net.WebClient).downloadString()` and `Invoke-WebRequest`. On Linux and macOS systems, a variety of utilities also exist, such as `curl`, `scp`, `sftp`, `tftp`, `rsync`, `finger`, and `wget`. (Citation: t1105_lolbas)

Name

Active Scanning

ID

T1595

Description

Adversaries may execute active reconnaissance scans to gather information that can be used during targeting. Active scans are those where the adversary probes victim infrastructure via network traffic, as opposed to other forms of reconnaissance that do not involve direct interaction. Adversaries may perform different forms of active scanning depending on what information they seek to gather. These scans can also be performed in

various ways, including using native features of network protocols such as ICMP.(Citation: Botnet Scan)(Citation: OWASP Fingerprinting) Information from these scans may reveal opportunities for other forms of reconnaissance (ex: [Search Open Websites/Domains](https://attack.mitre.org/techniques/T1593) or [Search Open Technical Databases](https://attack.mitre.org/techniques/T1596)), establishing operational resources (ex: [Develop Capabilities](https://attack.mitre.org/techniques/T1587) or [Obtain Capabilities](https://attack.mitre.org/techniques/T1588)), and/or initial access (ex: [External Remote Services](https://attack.mitre.org/techniques/T1133) or [Exploit Public-Facing Application](https://attack.mitre.org/techniques/T1190)).

Indicator

Name

9199814b65ec9529163b3e1a13efe54e4864383a

Pattern Type

yara

Pattern

rule CISA_10443863_01 : backdoor remote_access_trojan webshell exploitation information_gathering remote_access accesses_remote_machines anti_debugging captures_system_state_data controls_local_machine compromises_data_availability compromises_data_integrity fingerprints_host installs_other_components { meta: Author = "CISA Code & Media Analysis" Incident = "10443863" Date = "2023-05-11" Last_Modified = "20230522_1200" Actor = "n/a" Family = "n/a" Capabilities = "accesses-remote-machines anti-debugging captures-system-state-data controls-local-machine compromises-data-availability compromises-data-integrity fingerprints-host installs-other-components" Malware_Type = "backdoor remote-access-trojan webshell" Tool_Type = "exploitation information-gathering remote-access" Description = "Detects obfuscated and deobfuscated interactive PHP webshell samples" SHA256 = "ea98368f6ecb5281654a6a9e4c649ef9b53860f1ee32340145b61e0e42e1072a" strings: $e0 = { 65 76 61 6c } $e1 = { 72 6f 74 31 33 } $e2 = { 62 61 73 65 36 34 } $e3 = { 67 7a 69 6e 66 6c 61 74 65 } $e4 = { 73 68 65 6c 6c } $e5 = { 78 61 69 73 79 6e 64 69 63 61 74 65 } $e6 = { 54 75 62 61 67 75 73 4e 4d } $s0 = { 58 30 4d 42 31 33 } $s1 = { 74 75 6e 61 66 65 65 73 68 } $s2 = { 70 61 73 73 77 6f 72 64 } $s3 = { 6f 6e (63 | 43) 6c 69 63 6b 3d } $s4 = { 6a 61 76 61 73 63 72 69 70 74 3a 78 79 6e } condition: (6 of ($e*)) or (3 of ($s*)) }

Name

6ce087b904af8a01aae73ac77d81822ad41799f89a5d301dce45191c897012aa

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '6ce087b904af8a01aae73ac77d81822ad41799f89a5d301dce45191c897012aa']

Name

fa5867aeee505d1bbcaf70002e1dbd20af1e97c7

Pattern Type

yara

Pattern

rule CISA_10443863_02 : backdoor remote_access_trojan webshell exploitation information_gathering remote_access accesses_remote_machines anti_debugging captures_system_state_data controls_local_machine compromises_data_availability compromises_data_integrity fingerprints_host installs_other_components { meta: Author = "CISA Code & Media Analysis" Incident = "10443863" Date = "2023-05-11" Last_Modified = "20230522_1200" Actor = "n/a" Family = "n/a" Capabilities = "accesses-remote-machines anti-debugging captures-system-state-data controls-local-machine compromises-data-availability compromises-data-integrity fingerprints-host installs-other-components" Malware_Type = "backdoor remote-access-trojan webshell" Tool_Type = "exploitation information-gathering remote-access" Description = "Detects interactive ASP NET webshell samples" SHA256 = "ea98368f6ecb5281654a6a9e4c649ef9b53860f1ee32340145b61e0e42e1072a" strings: $s0 = { 3c 25 40 20 50 61 67 65 20 4c 61 6e 67 75 61 67 65 3d 22 43 23 22 } $s1 = { 62 61 73 65 36 34 (44 | 64) 65 63 6f 64 65 } $s2 = { 53 65 6c 65 63 74 20 2a 20 66 72 6f 6d 20 57 69 6e 33 32 5f 50 72 6f 63 65 73 73 } $s3 = { 53 45 4c 45 43 54 20 2a 20 46 52 4f 4d } $s4 = { 73 71 6c 63 6d 64 2e 65 78 65 } $s5 = { 63 6d 64 2e 65 78 65 } $s6 = { 49 49 53 20 56 65 72 73 69 6f 6e } $s7 = { 43 72 65 61 74 65 4e 6f 57 69 6e 64 6f 77 } condition: all of them }

Name

437eee0e5ab8db0f51e00ba421bbedf4581e6862

Pattern Type

yara

Pattern

rule CISA_10443863_03 : backdoor remote_access_trojan webshell exploitation information_gathering remote_access accesses_remote_machines anti_debugging captures_system_state_data controls_local_machine compromises_data_availability compromises_data_integrity fingerprints_host installs_other_components { meta: Author = "CISA Code & Media Analysis" Incident = "10443863" Date = "2023-05-16" Last_Modified = "20230605_1500" Actor = "n/a" Family = "n/a" Capabilities = "accesses-remote-machines anti-debugging captures-system-state-data controls-local-machine compromises-data-availability compromises-data-integrity fingerprints-host installs-other-components" Malware_Type = "backdoor remote-access-trojan webshell" Tool_Type = "exploitation information-gathering remote-access" Description = "Detects .NET DLL webshell samples" SHA256 = "b63c95300c8e36b5e6d3393da12931683796f88fd4601ba8364658b4d12ac05b" strings: $s0 = { 53 00 65 00 6c 00 65 00 63 00 74 00 20 00 2a 00 20 00 66 00 72 00 6f 00 6d 00 20 00 57 00 69 00 6e 00 33 00 32 00 5f 00 50 00 72 00 6f 00 63 00 65 00 73 00 73 } $s1 = { 62 61 73 65 36 34 (44 | 64) 65 63 6f 64 65 } $s2 = { 53 00 45 00 4c 00 45 00 43 00 54 00 20 00 2a 00 20 00 46 00 52 00 4f 00 4d } $s3 = { 49 00 49 00 53 00 20 00 41 00 50 00 50 00 50 00 4f 00 4f 00 4c } $s4 = { 4d 61 6e 61 67 65 6d 65 6e 74 4f 62 6a 65 63 74 } $s5 = { 43 72 65 61 74 65 4e 6f 57 69 6e 64 6f 77 } $s6 = { 73 71 6c 71 75 65 72 79 } condition: all of them }

Name

ea98368f6ecb5281654a6a9e4c649ef9b53860f1ee32340145b61e0e42e1072a

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'ea98368f6ecb5281654a6a9e4c649ef9b53860f1ee32340145b61e0e42e1072a']

Name

b63c95300c8e36b5e6d3393da12931683796f88fd4601ba8364658b4d12ac05b

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'b63c95300c8e36b5e6d3393da12931683796f88fd4601ba8364658b4d12ac05b']

StixFile

Value
6ce087b904af8a01aae73ac77d81822ad41799f89a5d301dce45191c897012aa
b63c95300c8e36b5e6d3393da12931683796f88fd4601ba8364658b4d12ac05b
ea98368f6ecb5281654a6a9e4c649ef9b53860f1ee32340145b61e0e42e1072a

External References

- https://www.cisa.gov/news-events/analysis-reports/ar23-166a
- https://otx.alienvault.com/pulse/648b7bb7a04269c57619fc06