# Here's a Simple Script to Detect the Stealthy Nation-State BPFDoor

**O** blog.qualys.com/vulnerabilities-threat-research/2022/08/01/heres-a-simple-script-to-detect-the-stealthy-nation-state-bpfdoor

#### Harshal Tupsamudre

#### August 1, 2022

```
int main(int argc, char *argv[])
{
        char hash[] = {0x6a, 0x75, 0x73, 0x74, 0x66, 0x6f, 0x72, 0x66, 0x75, 0x6e, 0x00}; // justforfun
        char hash2[]= {0x73, 0x6f, 0x63, 0x6b, 0x65, 0x74, 0x00}; // socket
        char *self[] = {
               "/sbin/udevd -d",
                "/sbin/mingetty /dev/tty7",
                "/usr/sbin/console-kit-daemon --no-daemon",
                "hald-addon-acpi: listening on acpi kernel interface /proc/acpi/event",
                "dbus-daemon --system",
                "hald-runner",
                "pickup -l -t fifo -u",
                "avahi-daemon: chroot helper",
                "/sbin/auditd -n",
                "/usr/lib/systemd/systemd-journald"
        };
```

In this blog, the Qualys Research Team explains the mechanics of a Linux malware variant named BPFdoor. We then demonstrate the efficacy of Qualys Custom Assessment and Remediation to detect it, and Qualys Multi-Vector EDR to protect against it.

BPFDoor is a Linux/Unix backdoor that allows threat actors to remotely connect to a Linux shell to gain complete access to a compromised device. It supports multiple protocols for communicating with a command & control server (C2) including TCP, UDP, and ICMP. It notably utilizes Berkeley Packet Filters (<u>BPF</u>) along with several other techniques to achieve these goals. BPF is a hooking function that allows a user-space program to attach a network filter onto any socket, and then allows or disallows certain types of data to come through that socket.

BPFDoor has been attributed to a Chinese threat actor group named Red Menshen (aka DecisiveArchitect), where the attackers have used it to gain stealthy remote access to compromised devices starting back in 2018 to the present day. Systems have been compromised across the US, South Korea, Hong Kong, Turkey, India, Vietnam, and Myanmar. Targets have included telecommunications, government, education, and logistics organizations. The group has been seen sending commands to BPFDoor victims via Virtual Private Servers (VPS) hosted at a well-known provider. In turn, these VPSs are administered via compromised routers based in Taiwan that the threat actor uses as VPN tunnels.

Target Geographies: Middle East, Asia

Target Sectors: Logistics, Education, Government

Malware Tools: Mangzamel, Gh0st, Gh0st, Metasploit, BPFDoor

#### Execution

The threat actor leverages a custom implant tracked by the name "JustForFun". When executed, the implant overwrites the process command line within the process environment by randomly selecting a new binary name from one of ten hard-coded options (shown in Figure 1). This masquerading technique is used to evade security solutions.

Figure 1: List of process names for Masquerading

The attacker interacts with the implant through the bash process to establish an interactive shell on a system. The command indicates the usage of Postfix queue manager (shown in Fig. 2).

Figure 2: Encoded shell and qmgr commands

Masquerading (Rename the process)

```
int set proc name(int argc, char **argv, char *new)
Ł
        size_t size = 0;
        int i;
        char *raw = NULL;
        char *last = NULL;
        argv0 = argv[0];
        for (i = 0; environ[i]; i++)
                size += strlen(environ[i]) + 1;
        raw = (char *) malloc(size);
        if (NULL == raw)
                return -1;
        for (i = 0; environ[i]; i++)
        {
                memcpy(raw, environ[i], strlen(environ[i]) + 1);
                environ[i] = raw;
                raw += strlen(environ[i]) + 1;
        last = argv[0];
        for (i = 0; i < argc; i++)</pre>
               last += strlen(argv[i]) + 1;
        for (i = 0; environ[i]; i++)
                last += strlen(environ[i]) + 1;
        memset(argv0, 0x00, last - argv0);
        strncpy(argv0, new, last - argv0);
        prctl(PR SET NAME, (unsigned long) new);
        return 0;
}
```

Figure 3: Code uses prctl to rename the malware process

The malware will rename itself using the prctl function with the argument PR\_SET\_NAME, and a random legitimate-looking name (Fig. 3). These names are hardcoded in the binary and vary between the samples.

### Timestomping

```
static void setup_time(char *file)
{
    struct timeval tv[2];
    tv[0].tv_sec = 1225394236;
    tv[0].tv_usec = 0;
    tv[1].tv_sec = 1225394236;
    tv[1].tv_usec = 0;
    utimes(file, tv);
}
```

Timestomping

The implant sets a fake time to timestomp the binary before deletion. A function dubbed set\_time was called to alter the access and modification timestamp of the binary using the utimes function (Fig. 4). The timestamp used was always set to Thursday, October 30, 2008 7:17:16 PM (GMT).

### PID File

The implant creates a zero-byte PID file at /var/run/haldrund.pid (Fig. 5). The file has two conditions:

- This file is deleted if the implant terminates normally,
- The file is not deleted, if there is a problem like hard shutdown or crash.

The implant will not resume if this file is present as it describes the running state for the backdoor.

```
pid_path[0] = 0x2f; pid_path[1] = 0x76; pid_path[2] = 0x61;
pid_path[3] = 0x72; pid_path[4] = 0x2f; pid_path[5] = 0x72;
pid_path[6] = 0x75; pid_path[7] = 0x6e; pid_path[8] = 0x2f;
pid_path[9] = 0x68; pid_path[10] = 0x61; pid_path[11] = 0x6c;
pid_path[12] = 0x64; pid_path[13] = 0x72; pid_path[14] = 0x75;
pid_path[15] = 0x6e; pid_path[16] = 0x64; pid_path[17] = 0x2e;
pid_path[18] = 0x70; pid_path[19] = 0x69; pid_path[20] = 0x64;
pid_path[21] = 0x00; // /var/run/haldrund.pid
```

Figure 5: Encoded command for creating PID file

# **BPFDoor Detection using Qualys Custom Assessment & Remediation**

Qualys Custom Assessment and Remediation can be leveraged to create and execute custom detection logics for zero-day threats. This cloud service supports multiple scripting languages including Perl, Shell, Python, Lua, PowerShell, and VBScript with no vendor-

specific syntax or restrictions. Select the language of your choice and start by leveraging outof-the-box scripts or creating your own scripts for custom detection, validation, and remediation.

We created the Shell script as part of our detection logic via the Qualys scripting service and executed it across the network.

Using this script, we are looking for packet sniffing processes under the entire process stack and checking if an existing process has opened a raw socket using the default Linux utility lsof. Refer the following screenshots of the script (Fig. 6) and its output (Fig. 7).

Qualys. Cloud Platfor	m		
$\leftarrow$ Update Script: bpf	door detection		
STEPS 3/3 Script Details 2 Add Script 3 Review & Confirm	Review and Confirm Basic Information Name: bpfdoor detection		
-	Description :	BPFDoor provides easy access to network packets and the ability to take actions via programs written based on custom filters before they ever reach a local firewall.	
	Platform :	LINUX	
	Type :	Shell	
	Category:	Data Collection	
	Severity :	5	
	Threshold :	300 seconds	
	Script		
	#!/bin/bash pid=\$(grep packet_recvms lsof -RPnI   grep SOCK_RAI if [[ \$? == 0 ]] then echo "Compromised" else echo "Not compromised" fi	sg /proc/*/stack   awk -F:' '{print \$1}'   grep -Po '\d+') W   grep IP   grep "\$pid" > /dev/null 2>&1	
	Cancel Previous	Update	

Figure 6: Script to detect BPFdoor

Qualys. Cloud Platform		Output of Asset: rhel8-2107-node01	
$\leftarrow$ View Details: Test-b		Execution Status: SUCCESS	
		The Script has following blacklisted commands in it. Please review if these commands are acceptable. The script may have changed post this job was run.	
<b>1</b> Total Asset		Black Listed Commands	
	JOB FINISHED ASSET TITLE	Isof	
STATUS	Jun 21, 2022 rhel8-2107-node01 01:09 AM 10.113.107.182	Script: bpfdoor detection	
SUCCESS 1		#//bin/bash pid=S(grep packet, recvmsg / proc/*/stack   awk -fr: '(print \$1)'   grep -Po `\d+') isor. #Phil [grep SOCK_RAW   grep IP   grep "Spid" > /dev/null 2>&1 if [[ \$7 == 0 ]] then echo "Compromised" echo "Not compromised" fi Output Compromised	

Figure 7: BPFdoor detection

# **BPFDoor Detection using Qualys Multi-Vector EDR**

Qualys Multi-Vector EDR, armed with YARA scanning techniques, detects the BPFdoor RAT with a threat score of 5/10 (Fig. 8).

C8346bf443b7b453f062740d8ae8d8d7ce879672810f429615 Path: /home/test/Downloads	36190359dcae3a	
Threat Details		
Threat Name Linux, Backdoor, BPFDoor	Category Backdoor	Score
incidem Description		
File		
File Action CREATED	File Type regularfile	File Extension
Tile Name dc8346bf443b7b453f062740d8ae8d8d7ce879672810f4296158f90359dcae3a	File Size 28.64 KB	Version -
created On Dec 31, 1979 12:00 AM	Modified 0n Dec 31, 1979 12:00 AM	Accessed On Dec 31, 1979 12:00 AM
roduct	Company -	Copyright -
home/test/Downloads	Full Path /home/test/Downloads/do8346bf443b7b453f0s2740d8ae8d8d7ce879672810f4296158f90359dca e3a	MD5 4574b9w820d22cd11d53wx2f1b56b045 G
SHA256 dc83466f4436754536062740d8ae8d8d7ce879672810f4296158f90359dcae3a 🌀 a duu	total 🗁	
Event		
D RTF_f71dfd74-7d5a-3ac7-86cf-aae4864296f7_23-6-2022	Object Type FILE	Event Collected Date Jun 23, 2022 03:43 PM

Figure 8: Qualys Multi-Vector EDR detection for BPFdoor

After execution, the binary masquerades its name by selecting from one of 10 names randomly:

/sbin/udevd -d /sbin/mingetty /dev/tty7 /usr/sbin/console-kit-daemon --no-daemon hald-addon-acpi: listening on acpi kernel interface /proc/acpi/event dbus-daemon --system hald-runner pickup -l -t fifo -u avahi-daemon: chroot helper /sbin/auditd -n /usr/lib/systemd/systemd-journald

The highlighted name was used during the execution. The names are made to look like common Linux system daemons. The implant overwrites the argv[0] value which is used by the Linux /proc filesystem to determine the command line and command name to show for each process. By doing this, when a run command like ps is executed, it shows the fake name.

The renamed binary is dropped to the /dev/shm directory and runs itself as /dev/shm/kdmtmpflush (Figs. 9 and 10). The masqueraded process with a "–init" flag tells itself to execute secondary clean-up operations and go resident.

Qualys. Cloud Platform	n				
vent Details:/dev/	/shm/kdmtmpflush			( Previous) (	
00E	Summary	Summary			
Event History Image Certificate	/dev/shm/kdmtmpflush			osLinux	
odules				Identification	
ie.	Threat Details			DNS Hostname node02-574	
	Thread Name	Category	Scare	FQDN node02-574	
				IPv4 10.113.107.196	
				1Pv6 fe80.0.0.0.19c7.ecc2.e52a.eet	
				Asset ID 25649556	
	Process			Activity	
	State RUNNING	Hana /devichen/destregfluch Dinoted true	nu han Alah Lain Kademangkush Unemene Koot	Lest User Login test	
	Argumenta			Last System Bo Jun 8, 2022 02:16 PM	
	/dev/shm/kdmtmpflushinit			Created On May 16, 2022 06:23 PM	
	ю 1287781			Last Checked In Jun 22, 2022 12:18 PM	
				Location	
	Event				
	ID RTP_02736442-a786-3df8-a9fd-15652a595d3a_22-6-2022	Object Type PROCESS	Event Collected Date Jun 22, 2022 12:20 PM		
				Lazardar verhanne Connected Proor: 12.112.127.136	
				Tage	

Figure 9: Qualys Multi-Vector EDR telemetry for detecting Masquerading

Qualys. Cloud Platform				
$\leftarrow$ Event Details:/dev/sh	m/kdmtmpflush			Yervious Next >
VIEW MODE Summary Event History Image	Process Tree			
Certificate Loaded Modules Process Tree	+	⊌⊿	Event Collected	2022 Jun 22, 2022 12:20 PM
			Object Type Process State	PROCESS
	O /home/less//D O Process 1 O /levionNar	renpfush 📀	Name Full Path Arguments Elevated Username ID Parent Process ID	/home/test/Downloads/sample/backdoor /home/test/Downloads/sample/bpfdoor.c /backdoor true root 1287772 /usr/bin/sudo
			Parent Process	

Figure 10: BPFdoor Process Tree

The implant creates a zero-byte PID file at /var/run/haldrund.pid (Fig. 11).

Qualys. Cloud Platform				
← Event Details:haldrur	id.pid			Vent >
VIEW MODE	Summary			Asset Details
Summary Event History Parent Process Process Tree	haldrund.pid Pedr: /ver/run			node02-574
Process Tree		Identification		
	Threat Details			DNS Hostname node02-574
	Threat Name	Category	Score	FQDN node02-574
	8	С.	0	IPv4 10.113.107.196
				IPv6 fe80:0:0:0:19c7.ecc2ie52a:ae6e
				Asset ID 25649556
	File			Activity
	File Action CREATED	File Type regularfile	File Extension pld	Last User Login test
	File Name	File Size	Version	Last System Bo Jun 8, 2022 02:15 PM
	heldrund pid	28.27 KB		Created On May 16, 2022 06:23 PM
	Created On Jun 22, 2022 11:53 AM	Modified On Jun 22, 2022 11:53 AM	Accessed On Jun 22, 2022 11:53 AM	Last Checked In Jun 22, 2022 02:18 PM
	Product	Company	Copyright	Location
	*		-	
	Path /var/run	Full Path /var/run/haldrund.pid	MD5 d41d8cd98f00b204e9800998ecf8427e	
	9425 9425 9300-4236-1-1-4949/Accesse/B924274414-6449693464495991578526855 G a.	Lostin unknown Dorrent (1113) 107-109		
	Event	77.4		
	ID RTF_d7840dec-0de6-3af1-9c72-473f6e5819e9_22-6-2022	Object Type FILE	Even Collected Date Jun 22, 2022 11:53 AM	

Figure 11: Creation of PID file by BPFdoor

As shown in figure 12, The original execution process deletes /dev/shm/kdmtmpflush with the following command:

/bin/rm -f /dev/sfm/kdmtmpflush

VIEW MODE Summary	Summary			
Evert Hatory Image Certificate Loaded Modules Process Tree	ر ن /usr/bin/rm			
	Threat Details Treat Name Linux_T1070_004_1	Category	Score 3	
	MITRE ATT&CK Technique(s)           TECHNIQUE ID         TECHNIQUE NAME           T1070.004         Indicator Removal on Host: File	Deletion		
	MITRE ATT&CK Tactic(s) Indite IB Indite Make TA0005 Defense Evaluer			
	Process Bate TERNINATED Agametes /bai/m-1/dev/ahm/kdmtmpflush 10 1287776	Nama Aaz/bistm Deettid true	Full Para Joar Diolom Usename root	

Figure 12: Deletion of /dev/shm/kdmtmpflush directory

# Conclusion

As with most remote access tools, BPFDoor is visible during the post-exploitation phase of an attack. It is expected that the authors behind BPFdoor will be upgrading its functionality over time, including different commands, processes, or files. This malware has a vast arsenal at its disposal. Therefore, we recommend that organizations have a robust EDR solution to both detect its signatures and adequately respond to the threat.

# MITRE ATT&CK Techniques

T1036.005- Masquerading: Match Legitimate Name or Location T1070.004- Indicator Removal on Host: File Deletion T1070.006- Indicator Removal on Host: Time Stomp T1059.004- Command and Scripting Interpreter: Unix Shell T1106- Native API T1548.001- Abuse Elevation Control Mechanism: Setuid and Setgid T1095- Non-Application Layer Protocol

# IoC (Indicators of Compromise)

### Hashes (SHA256)

07ecb1f2d9ffbd20a46cd36cd06b022db3cc8e45b1ecab62cd11f9ca7a26ab6d 1925e3cd8a1b0bba0d297830636cdb9ebf002698c8fa71e0063581204f4e8345 4c5cf8f977fc7c368a8e095700a44be36c8332462c0b1e41bff03238b2bf2a2d 591198c234416c6ccbcea6967963ca2ca0f17050be7eed1602198308d9127c78 599ae527f10ddb4625687748b7d3734ee51673b664f2e5d0346e64f85e185683 5b2a079690efb5f4e0944353dd883303ffd6bab4aad1f0c88b49a76ddcb28ee9

5faab159397964e630c4156f8852bcc6ee46df1cdd8be2a8d3f3d8e5980f3bb3 76bf736b25d5c9aaf6a84edd4e615796fffc338a893b49c120c0b4941ce37925 93f4262fce8c6b4f8e239c35a0679fbbbb722141b95a5f2af53a2bcafe4edd1c 96e906128095dead57fdc9ce8688bb889166b67c9a1b8fdb93d7cff7f3836bb9 97a546c7d08ad34dfab74c9c8a96986c54768c592a8dae521ddcf612a84fb8cc c796fc66b655f6107eacbe78a37f0e8a2926f01fecebd9e68a66f0e261f91276 c80bd1c4a796b4d3944a097e96f384c85687daeedcdcf05cc885c8c9b279b09c f47de978da1dbfc5e0f195745e3368d3ceef034e964817c66ba01396a1953d72 f8a5e735d6e79eb587954a371515a82a15883cf2eda9d7ddb8938b86e714ea27 fa0defdabd9fd43fe2ef1ec33574ea1af1290bd3d763fdb2bed443f2bd996d73 fd1b20ee5bd429046d3c04e9c675c41e9095bea70e0329bd32d7edd17ebaf68a 144526d30ae747982079d5d340d1ff116a7963aba2e3ed589e7ebc297ba0c1b3 fa0defdabd9fd43fe2ef1ec33574ea1af1290bd3d763fdb2bed443f2bd996d73 76bf736b25d5c9aaf6a84edd4e615796fffc338a893b49c120c0b4941ce37925 96e906128095dead57fdc9ce8688bb889166b67c9a1b8fdb93d7cff7f3836bb9 c80bd1c4a796b4d3944a097e96f384c85687daeedcdcf05cc885c8c9b279b09c f47de978da1dbfc5e0f195745e3368d3ceef034e964817c66ba01396a1953d72 07ecb1f2d9ffbd20a46cd36cd06b022db3cc8e45b1ecab62cd11f9ca7a26ab6d 4c5cf8f977fc7c368a8e095700a44be36c8332462c0b1e41bff03238b2bf2a2d 599ae527f10ddb4625687748b7d3734ee51673b664f2e5d0346e64f85e185683 5b2a079690efb5f4e0944353dd883303ffd6bab4aad1f0c88b49a76ddcb28ee9 5faab159397964e630c4156f8852bcc6ee46df1cdd8be2a8d3f3d8e5980f3bb3 93f4262fce8c6b4f8e239c35a0679fbbbb722141b95a5f2af53a2bcafe4edd1c 97a546c7d08ad34dfab74c9c8a96986c54768c592a8dae521ddcf612a84fb8cc c796fc66b655f6107eacbe78a37f0e8a2926f01fecebd9e68a66f0e261f91276 f8a5e735d6e79eb587954a371515a82a15883cf2eda9d7ddb8938b86e714ea27 fd1b20ee5bd429046d3c04e9c675c41e9095bea70e0329bd32d7edd17ebaf68a

#### Filenames

/dev/shm/kdmtmpflush /dev/shm/kdumpflush /dev/shm/kdumpdb /var/run/xinetd.lock /var/run/kdevrund.pid /var/run/haldrund.pid /var/run/syslogd.reboot

#### Process names

/sbin/udevd -d
/sbin/mingetty /dev/tty7
/usr/sbin/console-kit-daemon -no-daemon
hald-addon-acpi: listening on acpi kernel interface /proc/acpi/event
dbus-daemon -system
hald-runner
pickup -l -t fifo -u
avahi-daemon: chroot helper
/sbin/auditd -n
/usr/lib/systemd/systemd-journald
/usr/libexec/postfix/master
qmgr -l -t fifo -u

#### **Contributors:**

Viren Chaudhary (Senior Engineer, Threat Research, Qualys) Mukesh Choudhary (Compliance Research Analyst, Qualys) Lavish Jhamb (Solutions Architect, Compliance Solutions, Qualys) Mohd Anas Khan (Compliance Research Analyst, Qualys)