ADVISORY FOR ICM SECTOR

Earth Bluecrow Deploys BPFDoor Malware Across Telcos in Asia

Original report published on: April 14, 2025[1]

Executive Summary

On 14 April 2025, Trend Micro identified a new BPFDoor malware controller targeting Linux servers in the telecommunication, finance, and retail sectors across Asia and the Middle East. BPFDoor leverages Berkeley Packet Filtering (BPF) at the kernel level to create a stealthy backdoor, evading traditional security tools for long-term espionage operations.

Background

PwC first identified Red Menshen (later named Earth Bluecrow by Trend Micro) in 2021 and attributed BPFDoor malware (in linux) to them, as well as designated them as an APT group. This group conducted operations across South Korea, Hong Kong, Myanmar, Malaysia and Egypt in 2024.

The initial entry point remains unknown as the investigations into the 2024 incidents are still ongoing. The malware resided in various linux server paths including '/tmp/zabbix_agent.log', '/bin/vmtoolsdsrv', and '/etc/sysconfig/rhn/rhnsd.conf'.

BPFDoor uses Berkeley Packet Filtering (BPF)[1] to inspect network packets for 'magic bytes' - predefined byte sequences that trigger specific backdoor actions. Before sending these packets with 'magic bytes' the controller must first authenticate using a password. After verification, the controller commands the compromised machine to execute specific actions based on command-line options.

The malware performs three actions: (1) open a reverse shell, (2) redirect new connections to a shell on a specific port, and (3) confirm the backdoor's active status. BPFDoor supports multiple protocols (TCP, UDP, and ICMP), changes process names, operates without listening ports, and disables command logging by running the command (HISTFILE=/dev/null) to evade detection.

Detection and Mitigation

IMDA recommends organisations perform continual testing and validation of security controls to ensure detection and prevention of the BPFDoor malware identified in this advisory:

- Scan for Indicators of Compromise to detect threat activities (Annex A).
- Refer to the MITRE ATT&CK techniques (Annex B) in this advisory:
  - Create, test and validate detection rules against the threat behaviours.
  - Validate and deny/disable processes, ports and protocols that have no business need.

[1] Berkeley Packet Filtering is a kernel-level mechanism that runs lightweight programs to inspect and filter network packets based on criteria such as protocol, port, or specific byte patterns – deciding whether to drop them, pass them to user-space tools like 'tcpdump', or let them continue through the networking stack.

- Review and test the BPFDoor YARA rule (Annex C) for file scanning in a controlled environment to ensure accuracy and stability, then deploy it across relevant systems and security tools to detect malicious files or processes.
- Leverage Network Detection and Response (NDR) solutions capable of analysing non-standard protocol behaviour and inspecting raw packets for signs of 'magic bytes' typical of BPFdoor C2 activity.
- Detect the use of environment variables like 'HISTFILE=/dev/null' that suppress shell command logging.
- Utilise kernel and BPF activity monitoring tools to monitor for unusual BPF filter activity, raw socket creation, or anomalous kernel interactions.
- IMDA encourages organisations to conduct thorough analysis to identify potential risks and assess their potential impact prior to deploying defensive measures.

Annex A - Indicators of Compromise

Type	Value	Comment
SHA1	6227cb77cb4ab1d066eebf14e825dbc0a0a7f1e9	BPFDoor Controller
SHA1	64171d46c8290c5cd88e0fbce9e23dcecbe20865	
SHA1	65e4d507b1de3a1e4820e4c81808fdfd7e238e10	
SHA1	9bb8977cd5fc7be484286be8124154ab8a608d96	
SHA1	02aebc3762e766be0ac24ef57a135398344a8f7e	BPFDoor Malware
SHA1	04aa241c574f0a7ec93ba5d27807d8e78467f21e	
SHA1	16f94f0df6003f1566b2108f55e247f60a316185	
SHA1	1db21dbf41de5de3686195b839e74dc56d542974	
SHA1	28765121730d419e8656fb8d618b2068408fe5ae	
SHA1	291af8adf6fa078692d0bf5e0d9d00c376bb3fff	
SHA1	316ac8215095a24429632849407311b18a16e0cf	
SHA1	351febd645c66a3c9a79253d0aefcce8ff77054c	
SHA1	3771319be1c8883610c65977811e93b0bdaddf6f	
SHA1	4181bc848a1cd32911a83e02feac9b8abbd69ae2	
SHA1	43b4dbc71ada99a7b8a8d6d0490ba5526a34f9a0	
SHA1	4b1ea5e7b28eb110d8741b76d34f7dbae6f13b79	
SHA1	4c89beab00e3119cf516d6f98d364a5d99232181	
SHA1	5ddcbe4b591293f7b34fc0ef65db6248bcc67eb6	
SHA1	64171d46c8290c5cd88e0fbce9e23dcecbe20865	
SHA1	7ab39d7aad49abb0f626383ed776fe20a3b4c8f3	
SHA1	88075bbc34655d1fa2a750f3bbdee38214974009	
SHA1	8b5844fcbf6af23bc0b410fc180e7e6bdd4f35c7	
SHA1	8ebe1a71af1061d9e943bdff46c5ed954d8c9348	
SHA1	937f6068df4e091cf92d50afb3c6b7cad1de6230	
SHA1	9bdea37835cdb7f0b291891386af28184ac85f79	
SHA1	a6b66b8b7eae2969fd7237888d30766baa1b2274	
SHA1	be47b0c2fb328a338874f6efbe8305ddb74f6a48	
SHA1	c38fc109e31c9d67a1efc6cb767f826b7e46fb19	
SHA1	c54e214810ca7042d013845076b0360bdd7132b2	
SHA1	c83651d7706efa8c115f2a0edb07f863f4e79ce5	
SHA1	d53b7d0030a095a3ffa4b67d13de82d08adda248	
SHA1	d61bf187c4cd3f9953b567b3ad320b9ecde1c347	
SHA1	d9037e0de902e6f7b6c5f1b3269ba482f5e67c8a	
SHA1	dc94eaa39e11f2ca7739d2cfded9eec1967f33ee	
SHA1	dd8db29e90c6b52ee3d2723cc168cf33ee0bb521	
SHA1	e17ddb6515f2d399552245191f98458b68fece7b	
SHA1	ef03b84048193a158ecf1f7033ab0cc8869dd2a5	
SHA1	f61589b1f86d8692964a6bd3e96ddadbe22994eb	
SHA1	fb488cdfd2e475f0d5cbecfe11e9bab2241f9d50	
SHA256	c7f693f7f85b01a8c0e561bd369845f40bff423b0743c7aa0f4c323d9133b5d4	hpasmmld
SHA256	3f6f108db37d18519f47c5e4182e5e33cc795564f286ae770aa03372133d15c4	smartadm

SHA256	95fd8a70c4b18a9a669fec6eb82dac0ba6a9236ac42a5ecde270330b66f51595	hald-addon-volume
SHA256	aa779e83ff5271d3f2d270eaed16751a109eb722fca61465d86317e03bbf49e4	dbus-srv-bin.txt
SHA256	925ec4e617adc81d6fcee60876f6b878e0313a11f25526179716a90c3b743173	dbus-srv
SHA256	29564c19a15b06dd5be2a73d7543288f5b4e9e6668bbd5e48d3093fb6ddf1fdb	inode262394
SHA256	be7d952d37812b7482c1d770433a499372fde7254981ce2e8e974a67f6a088b5	dbus-srv
SHA256	027b1fed1b8213b86d8faebf51879ccc9b1afec7176e31354fbac695e8daf416	dbus-srv
SHA256	a2ea82b3f5be30916c4a00a7759aa6ec1ae6ddadc4d82b3481640d8f6a325d59	dbus-srv
SHA256	e04586672874685b019e9120fcd1509d68af6f9bc513e739575fc73edefd511d	File_in_Inode_#1900667
SHA256	adfdd11d69f4e971c87ca5b2073682d90118c0b3a3a9f5fbbda872ab1fb335c6	gm
SHA256	7c39f3c3120e35b8ab89181f191f01e2556ca558475a2803cb1f02c05c830423	rad
IPv4	165.232.174[.]130	Attack IP

Annex B - MITRE ATT&CK Tactics and Techniques

Tactic	Technique ID	Remark
Execution	T1059.004 – Command and Scripting Interpreter: Unix Shell	Kernel-level reverse shell initiated via BPF-based “magic packet” filters, enabling remote command execution.
Defence Evasion	T1070 – Indicator Removal on Host	Environment variables MYSQL_HISTFILE=/dev/null and HISTFILE=/dev/null are set to suppress shell command logging.
	T1036.004 – Masquerading: Masquerade Task or Service	Changes process name to avoid detection
Lateral Movement	T1021 – Remote Services	Reverse shell (TCP/UDP/ICMP) leveraged as a pivot point to access and laterally move to additional systems within the network.
Command and Control	T1095 – Non-Application Layer Protocol	C2 communication established over raw TCP, UDP, or ICMP packets, using “magic bytes” to maintain stealth and evade traditional application-layer inspection.

Annex C – ELF_BPFDoor YARA Rule

Reference: KISA KrCERT - BPFDoor Detection Methods[4], dated 12 May 2025, Paragraph 9, BPFDoor YARA Rule.

```
rule ELF_BPFDoor
{
    meta:
        description = "BPFDoor Malware(ELF) Detection rule"
        author = "KrCERT/CC Threat Hunting Analysis Team"
        date = "2025-05-06"
        hash = "c7f693f7f85b01a8c0e561bd369845f40bff423b0743c7aa0f4c323d9133b5d4"
        hash = "3f6f108db37d18519f47c5e4182e5e33cc795564f286ae770aa03372133d15c4"
        hash = "95fd8a70c4b18a9a669fec6eb82dac0ba6a9236ac42a5ecde270330b66f51595"
        hash = "aa779e83ff5271d3f2d270eaed16751a109eb722fca61465d86317e03bbf49e4"
        ver = "1.1"

    strings:
        $pty_1 = "/dev/ptm"
        $pty_2 = "ptem"
        $pty_3 = "ldterm"
        $pty_4 = "ttcompat"

    // for ( i = 0xA597; i <= 0xA97E; ++i )
    $bind_port_1  = {C7 45 ?? 97 A5 00 00 EB}
    $bind_port_2_1  = {83 45 ?? 01 81 7D ?? 7E A9 00 00 7E}
    $bind_port_2_2  = {81 7D ?? 7E A9 00 00 7E}

    // v1.1 : added new pattern "PS1=[\\u@\\h : "
    $ps1 = {C6 85 ?? ?? FF FF 50 C6 85 ?? ?? FF FF 53 C6 85 ?? ?? FF FF 31 C6 85 ?? ?? FF FF 3D C6
85 ?? ?? FF FF 5B C6 85 ?? ?? FF FF 5C C6 85 ?? ?? FF FF 75 C6 85 ?? ?? FF FF 40 C6 85 ?? ?? FF FF
5C C6 85 ?? ?? FF FF 68}

    // v1.1 : bpf filter modify (dbus, hald, trend-B, Trend-E)
    $bpf_filter_1 = {B1 00 00 00 0E 00 00 00 48 00 00 00 16 00 00 00 15 00 ?? ?? ?? ?? 00 00 15 00
00 07 01 00 00 00 28 00 00 00 14 00 00 00 45 00 (11|1A) 00 FF 1F 00 00}
    $bpf_filter_2 = {07 00 00 00 00 00 00 00 (40|48) 00 00 00 00 00 00 00 02 00 00 00 (00|01|03)
01 00 00 00 00 00 00 00 ?? ?? ?? ?? 02 00 00 00 (01|02|04) 00 00 00 61 00 00 00 (01|02|04) 00
00 00}
    $bpf_filter_3 = {07 00 00 00 00 00 00 00 40 00 00 00 0E 00 00 00 15 00 00 01 39 39 39 39 06 00
00 00 FF FF 00 00 06 00 00 00 00 00 00 00}
    $bpf_filter_4 = {48 00 00 00 10 00 00 00 15 00 01 00 BB 01 00 00 15 00 00 01 16 00 00 00 06 00
00 00 00 00 04 00 06 00 00 00 00 00 00 00}

    // I5*AYbs@LdaWbsO
    $md5_salt = {C6 45 ?? 49 C6 45 ?? 35 C6 45 ?? 2A C6 45 ?? 41 C6 45 ?? 59 C6 45 ?? 62 C6 45 ??
73 C6 45 ?? 40 C6 45 ?? 4C C6 45 ?? 64 C6 45 ?? 61 C6 45 ?? 57 C6 45 ?? 62 C6 45 ?? 73 C6 45 ??
4F}



    // qmgr -l -t fifo
```

```
    $pname = {C6 ?? ?? ?? FF FF 71 C6 ?? ?? ?? FF FF 6D C6 ?? ?? ?? FF FF 67 C6 ?? ?? ?? FF FF 72
C6 ?? ?? ?? FF FF 20 C6 ?? ?? ?? FF FF 2D C6 ?? ?? ?? FF FF 6C C6 ?? ?? ?? FF FF 20 C6 ?? ?? ?? FF FF
2D C6 ?? ?? ?? FF FF 74 C6 ?? ?? ?? FF FF 20 C6 ?? ?? ?? FF FF 66 C6 ?? ?? ?? FF FF 69 C6 ?? ?? ??
FF FF 66 C6 ?? ?? ?? FF FF 6F}

    $solaris_1 = "(udp[8:2]=0x7255) or (icmp[8:2]=0x7255)"
    $solaris_2 = "HISTFILE=/dev/null"
    $solaris_3 = "MYSQL_HISTFILE=/dev/null"

    condition:
    uint32(0) == 0x464C457F and
    (
        (all of ($pty*) and (2 of ($bind_port*) or $ps1)) or // v1.1
        ((all of ($pty*) or $ps1) and 2 of ($bind_port*)) or // v1.1
        (1 of ($bpf_filter*)) or
        ($md5_salt or $pname) or
        (all of ($solaris*))
    )
}


rule ELF_BPFDoor_Controller
{
    meta:
        description = "BPFDoor Malware(ELF) Controller Detection rule"
        author = "KrCERT/CC Threat Hunting Analysis Team"
        date = "2025-05-06"
        hash = "93f4262fce8c6b4f8e239c35a0679fbbbb722141b95a5f2af53a2bcafe4edd1c"
        hash = "1925e3cd8a1b0bba0d297830636cdb9ebf002698c8fa71e0063581204f4e8345"
        hash = "591198c234416c6ccbcea6967963ca2ca0f17050be7eed1602198308d9127c78"
        ver = "1.1"

    strings:
        // v1.1 : added new pattern
        $pname = /\/usr\/sbin\/[a-z]{2,8}/
        $str_1 = "[-] option requires an argument -- d"
        $str_2 = ":h:d:l:s:b:t:"

        // mov cs:magic_flag, 5571h
        $magic = {C7 05 ?? ?? ?? 00 71 55 00 00}

        $env_1 = "export TERM=vt100"
        $env_2 = "export MYSQL_HISTFILE=/dev/null"
        $env_3 = "export HISTFILE=/dev/null"
        $env_4 = "unset PROMPT_COMMAND"
        $env_5 = "export HISTSIZE=100"

    condition:
        uint32(0) == 0x464C457F and
        ($pname or 1 of ($str*)) and $magic and (all of ($env*))
}
```

References

1. BPFDoor's Hidden Controller Used Against Asia, Middle East Targets
2. New BPFDoor Controller Enables Stealthy Lateral Movement in Linux Server Attacks
3. AhnLab Detection Information on BPFDoor Exploited in Recent Hacking Attacks and KISA Hash Notice
4. KISA KrCERT: BPFDoor Detection Methods