

SNEAKY PATCH INVESTIGATION - PRE-REPORT - TIER 1 SECURITY ANALYST L1

INFORMATION:

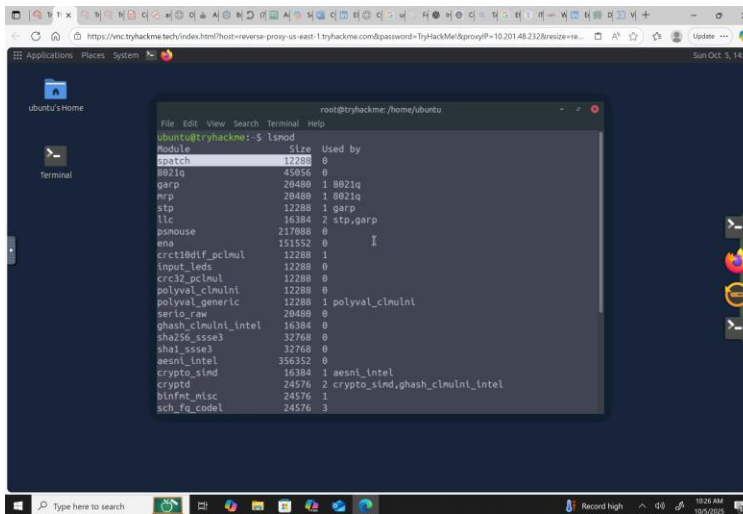
A HIGH-VALUE SYSTEM HAS BEEN COMPROMISED. SUSPICIOUS ACTIVITIES WITHIN THE KERNEL HAVE BEEN DETECTED. TRADITIONAL DETECTION TOOLS HAVE FAILED & THE INTRUDER HAS ESTABLISHED DEEO PERSISTENCE. MY TASK IS TO INVESTIGATE A LIVE SYSTEM SUSPECTED OF RUNNING A KERNEL-BACKDOOR.

SCOPE:

INVESTIGATION WAS CONDUCTED ON A LIVE SYSTEM WITH LIMITED ACCESS. NO EXTERNAL FORENSIC TOOLS WERE USED; ALL ANALYSIS WAS PERFORMED USING BASH COMMANDS FOR MODULE INSPECTION, LOG REVIEW, & SYSTEM EXPLORATION. CYBERCHEF WAS USED AS A SUPPLEMENTARY TOOL TO DECODE POTENTIAL ENCODED STRINGS FOUND IN THE MODULE.

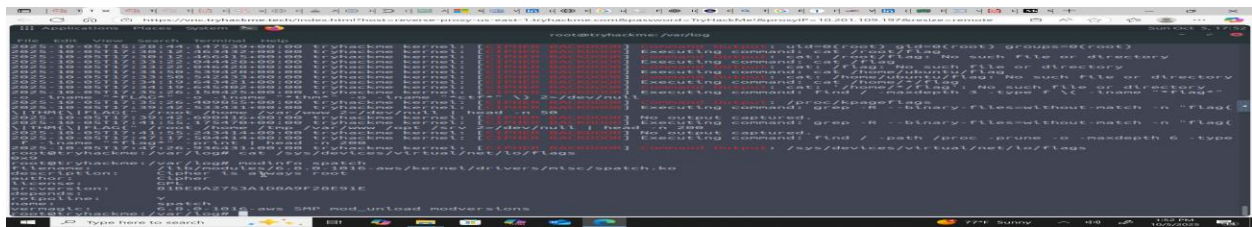
INVESTIGATION:

THE INVESTIGATION BEGAN WITH 'lsmod' TO IDENTIFY LOADED KERNEL MODULES. A NON-STANDARD MODULE NAMED SPATCH WITH SIZE 12288 WAS DETECTED, INDICATING POTENTIAL MALICIOUS ACTIVITY.

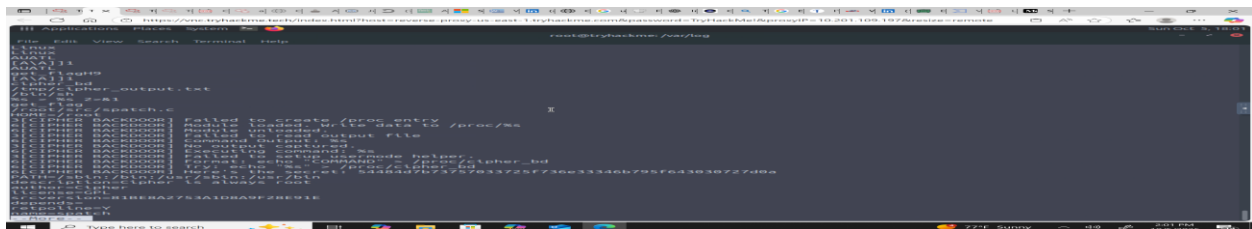


```
root@tryhackme: /home/ubuntu
lsmod
Module      Size  Used by
spatch      12288  0
8021q        45056  0
garp         20480  1 8021q
mrp          20480  1 8021q
stp          12288  1 garp
llic         16384  2 stp,garp
psnouse      217600  0
ena          151552  0
crc18dif_pci 12288  1
input_leds   12288  0
crc32_pci    12288  0
polyval_clmul 12288  0
polyval_generic 12288  1 polyval_clmul
serio_raw    20480  0
ghash_clmulni_intel 16384  0
sha256_ssse3 32768  0
sha1_ssse3   32768  0
aesni_intel 356352  0
crypto_simd 16384  1 aesni_intel
cryptd       24576  2 crypto_simd,ghash_clmulni_intel
binfmt_misc  24576  1
sch_fq_codel 24576  3
```

'modinfo spatch' WAS RUN TO IDENTIFY MODULE METADATA, INCLUDING FILE LOCATION & VERIFICATION STATUS, WHICH CONFIRMED IT WAS UNSIGNED & OUT-OF-TREE.



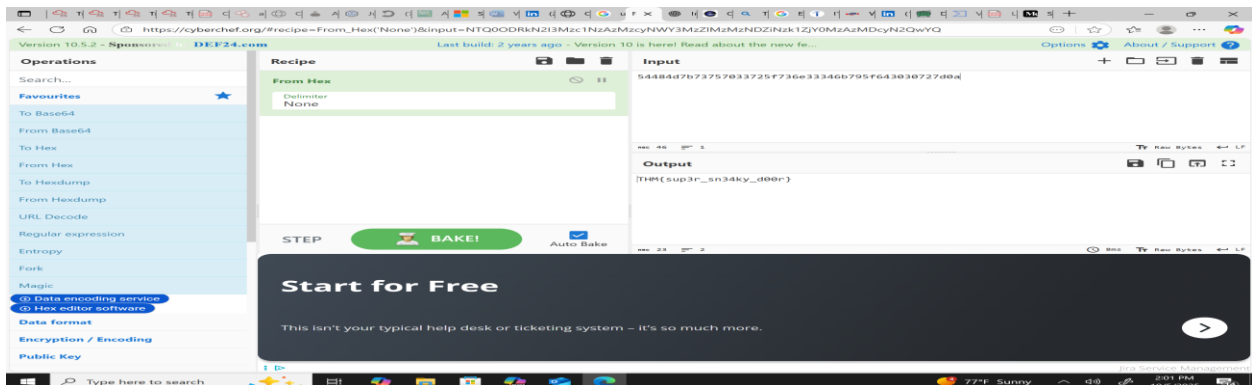
THE MODULE BINARY WAS EXAMINED USING 'strings /lib/modules/6.8.0-1016-aws/kernel/drivers/misc/spatch.ko', REVEALING AN ENCODED STRING LABELED 'HERE'S THE SECRET' SUGGESTIVE OF BASE64 OR HEX DATA.



LOG FILES (KERN.LOG & SYSLOG) WERE REVIEWED & CONFIRMED THAT THE MODULE CREATED A PROCFS CONTROL NODE (/PROC/CIPHER_BD) & EXECUTED ROOT-LEVEL COMMANDS VIA CRON AT REBOOT.



THE FLAG ASSOCIATED WITH THE EXERCISE WAS CAPTURED: THM{sup3r_sn34ky_d00r}.



CONCLUSION:

THE HOST TRYHACKME IS RUNNING AN UNSIGNED, OUT-OF-TREE KERNEL MODULE NAMED SPATCH THAT IMPLEMENTS A PROCFS CONTROL CHANNEL (/PROC/CIPHER_BD) & IS BEING PERSISTENTLY RELOADED VIA A ROOT@REBOOT CRON JOB. KERNEL LOGS CONFIRM THE MODULE LOADS, ACCEPTS COMMANDS VIA /PROC/CIPHER_BD, EXECUTES THEM IN KERNEL CONTEXT, & RETURNS OUTPUT — DEMONSTRATED BY

THE ID COMMAND & ITS OUTPUT IN KERN.LOG. THIS CONSTITUTES KERNEL-LEVEL CODE EXECUTION WITH ROOT PRIVILEGE & PERSISTENT BACKDOOR CAPABILITY.

KEY FINDINGS:

SPATCH IS UNSIGNED / OUT-OF-TREE → KERNEL TAINT RECORDED: “MODULE VERIFICATION FAILED ... TAINTING KERNEL.” PERSISTENCE MECHANISM: ROOT CRONTAB ENTRY @REBOOT ... DEPMOD -A && MODPROBE SPATCH && ECHO "ID" > /PROC/CIPHER_BD. CONTROL CHANNEL: MODULE CREATES /PROC/CIPHER_BD; WRITING COMMANDS TO IT CAUSES THE MODULE TO EXECUTE THEM; LOGS SHOW EXECUTION & OUTPUT. EVIDENCE OF OPERATOR ARTIFACTS DISCOVERED IN THE MODULE STRINGS INCLUDING AN EMBEDDED MARKER & THE FLAG THM{SUP3R_SN34KY_D00R} (EXERCISE ARTIFACT). NO RELEVANT INSMOD/MODPROBE ENTRIES FOUND IN AUTH LOGS — CRON IS THE INSTALL/RELOAD VECTOR. THE MODULE EXECUTED ID AS ROOT & LOGGED UID=0(ROOT) ... — PROVING KERNEL-LEVEL COMMAND EXECUTION.

IMPACT:

CRITICAL — FULL HOST COMPROMISE. A KERNEL MODULE CAN BYPASS USERLAND CONTROLS, HIDE ARTIFACTS, PERSIST ACROSS REBOOTS, & EXECUTE ARBITRARY PRIVILEGED OPERATIONS. IN A REAL PRODUCTION ENVIRONMENT THIS EQUALS COMPLETE LOSS OF HOST INTEGRITY & TRUST.

CONFIDENCE:

HIGH — CORROBORATED BY KERNEL LOGS (KERN.LOG), CRON PERSISTENCE ENTRY, PROCFS INTERACTION, & MODULE BINARY STRINGS/BEHAVIOR. ACTIONS TAKEN / EVIDENCE PRESERVED (RECOMMENDED & PERFORMED): CAPTURED KERN.LOG, DMESG/KERNEL JOURNAL ENTRIES, ROOT CRONTAB, & ANY LOCATED SPATCH.KO INTO A FORENSIC FOLDER (/TMP/FORENSIC) & RECORDED CHECKSUMS. DOCUMENTED THE CRON ENTRY & KERNEL LOG LINES SHOWING MODULE LOAD, COMMAND EXECUTION, & COMMAND OUTPUT. (SIMULATION) FLAG RETRIEVED & NOTED AS EVIDENCE OF EXERCISE COMPLETION.

RECOMMENDATIONS:

TREAT HOST AS FULLY COMPROMISED. IN PRODUCTION: ISOLATE FROM NETWORK IMMEDIATELY. FORENSIC COLLECTION: CAPTURE MEMORY IMAGE, FULL DISK IMAGE, ALL LOGS, & COPIES OF SPATCH.KO FOR OFFLINE REVERSE ENGINEERING. PRESERVE TIMESTAMPS & HASHES. OFFLINE ANALYSIS: REVERSE ENGINEER SPATCH.KO TO ENUMERATE CAPABILITIES (PERSISTENCE, C2, FILESYSTEM/NETWORK HOOKS, SYSCALL HOOKING). DO NOT RUN THE MODULE ON PRODUCTION SYSTEMS. REMEDIATION: REBUILD THE HOST FROM KNOWN-GOOD MEDIA AFTER FORENSIC CAPTURE. AVOID IN-PLACE CLEANUP UNLESS CONDUCTED BY EXPERIENCED RESPONDERS. CREDENTIAL & KEY ROTATION: ASSUME CREDENTIALS ARE COMPROMISED; ROTATE ROOT & SERVICE KEYS/SECRETS. SCOPE & HUNT: SEARCH OTHER SYSTEMS FOR THE SAME CRON ENTRY, SPATCH MODULE, /PROC/CIPHER_BD, OR ANY IOCs FOUND IN MODULE STRINGS. REVIEW NETWORK LOGS FOR SUSPICIOUS EGRESS. HARDENING: ENABLE MODULE SIGNATURE ENFORCEMENT WHERE POSSIBLE, AUDIT CRON/BOOT TASKS, & ENFORCE FILE INTEGRITY MONITORING ON /LIB/MODULES & /ETC/CRON*. REPORT & ESCALATE: ESCALATE TO SOC/TRIAGE & FOLLOW INCIDENT RESPONSE PLAYBOOK FOR KERNEL COMPROMISE.

FINAL ASSESSMENT:

THIS INVESTIGATION CONFIRMS AN INTENTIONAL KERNEL-LEVEL BACKDOOR WAS INSTALLED & PERSISTED VIA CRON, PROVIDING A PRIVILEGED COMMAND EXECUTION CHANNEL. THE EXERCISE DEMONSTRATES THE INDICATORS & ARTIFACTS REQUIRED FOR REAL INCIDENT RESPONSE. IN PRODUCTION, IMMEDIATE ISOLATION, FULL FORENSIC CONTAINMENT, & REBUILD ARE MANDATORY.