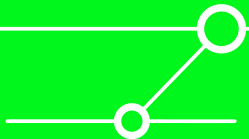


Hacking VoIP for Fun and Profit

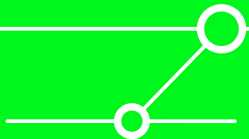
Roland Fiege, Geschäftsführer

ERNW GmbH

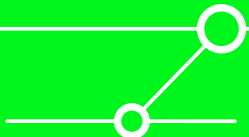


Agenda

- Kurzvorstellung ERNW
- Grundlagen / Begriffsklärung VoIP
- Sicherheitsziele
- Kleine Historie der VoIP-Vulnerabilities....
- Vorstellung einer neuen Sicherheitslücke in einer SIP-Bibliothek
- Wie werden solche Lücken entdeckt und publiziert?
- Potentielle Auswirkungen der Schwachstelle
- Was lernen wir daraus?

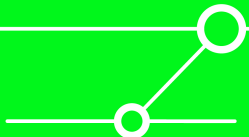


- Gegründet Sommer 2001
- Sitz in Heidelberg, Deutschland
- unabhängiger IT-Security-Dienstleister
- Aktuell elf Mitarbeiter
- Schwerpunkte:
 - Security Management
 - Audit/Revision (BS7799/ISO27001)
 - Penetrations-Tests
 - Risiko-Bewertung & -Management
 - Security Research
- Kunden: Industrie, Banken, Behörden, Provider



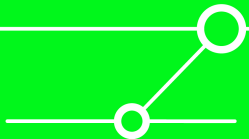
Grundlagen und Begriffe VoIP

- Der Transport von ‚paketierte‘ Telefonverbindungen über IP-Netzwerke. Meist (aber nicht nur) Audio-Daten, teilweise über bestimmte Gateways direkt aus dem *PSTN (Public Switched Telephone Network)*).
- Kann innerhalb von abgeschlossenen Netzeinheiten stattfinden (=> dann spricht man oft von *VoIP*) oder über das Internet (=> Terminus ist hier üblicherweise *IP Telephony*)
- Ich verwende im Vortrag beide Begriffe weitgehend austauschbar.
- VoIP ist kein Protokoll oder Standard
Eher ein Sammelbegriff für verschiedene Technologien.



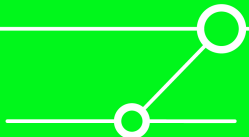
Bestandteile typischer VoIP-Szenarios

- Protokolle (Transport, Signalisierung, Management/Infrastruktur)
- Komponenten (dedizierte VoIP-Devices, Kopplungs-Geräte)
- Verbindungen (öffentliche, nicht-öffentliche Strecken)
- Endgeräte (Hardphones, Softphones, ...)
- User („Faktor Mensch“)

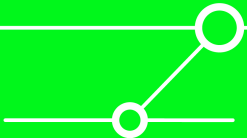
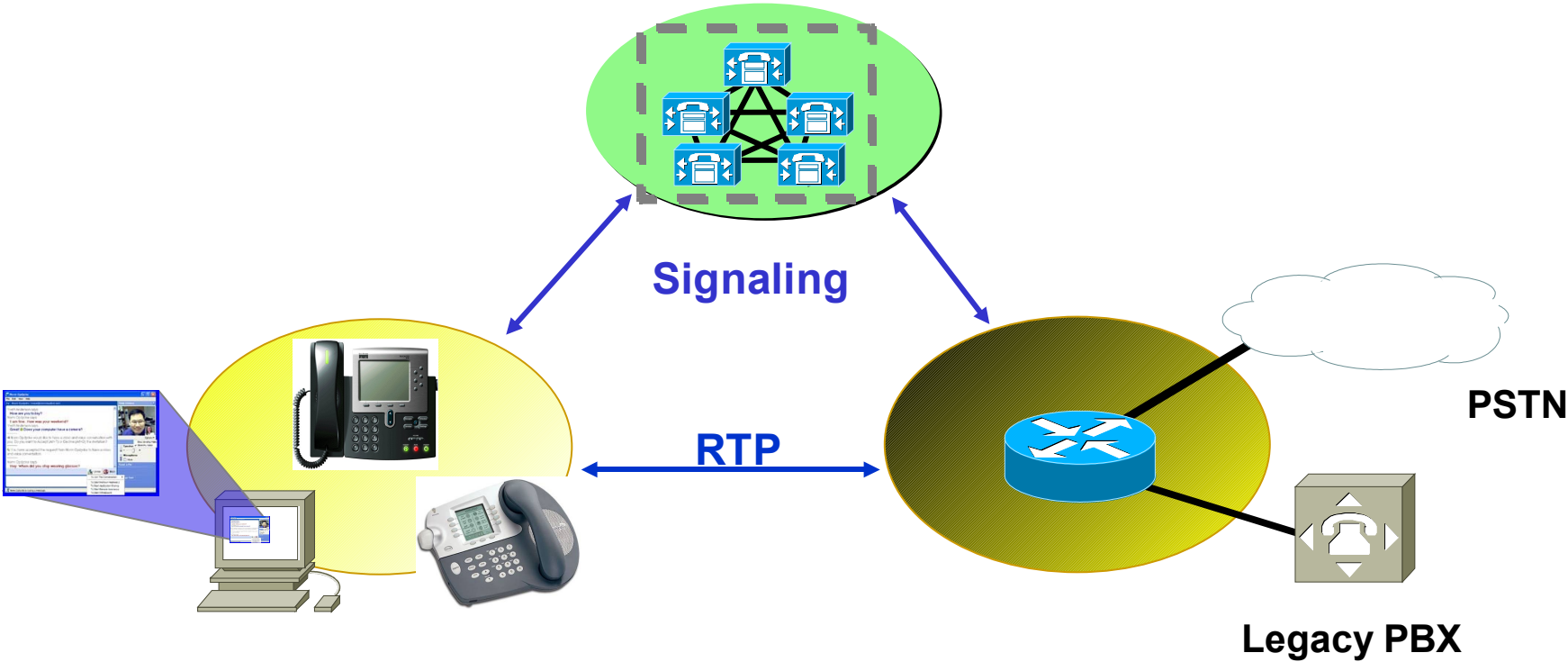


Protokolle

- Transport (von Nutzdaten):
Real-Time Transport Protocol [RTP, RFC 1889]
+ ggf. *RTP Control Protocol* [RTCP]
- Signalisierung (*Location of Users, Session Setup & Negotiation* etc.):
 - *H.323* [Urheber: ITU]
 - *Session Initiation Protocol* [SIP, RFC 2543/3261]
 - *Skinny/SCCP* [Cisco-proprietär]
 - *Media Gateway Control Protocol* [MGCP, RFC 2705]
 - *Megaco* [RFC 3015, ITU-T H.248]
- Management/Infrastruktur [DNS, SNMP et.al.]

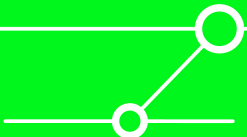


Zusammenwirken der Protokolle



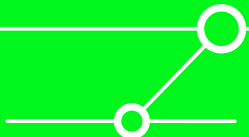
Komponenten

- Dedizierte VoIP-Devices, e.g.
 - H.323 Gatekeeper, Multipoint Controller
 - SIP Proxy, Redirector
 - Voice-Mail Server
 - TRIP Location Server
- Kopplungskomponenten zu anderen Netzen, z.B. GGSN/SGSN zu GPRS/UMTS-Netzen
- Infrastruktur-Komponenten (etwa Backbone Router)



Endgeräte

- Hardphones: dedizierte Telefone mit VoIP-Fähigkeiten
- Softphones: Software-Applikationen, die auf PCs laufen und VoIP-Funktionalität zur Verfügung stellen.
- Legacy Devices: traditionelle Telefon-/Fax-Geräte, die über geeignete Schnittstellen (etwa Cisco NM-2V+VIC-2FXS) in VoIP-Netze integriert werden.

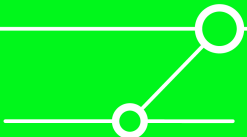


Sicherheitsziele

- Verfügbarkeit
- Vertraulichkeit
- Integrität
- Authentizität
- Non-Repudiation
- Einhaltung gesetzlicher Bestimmungen
 - Datenschutz
 - *Lawful Interception*

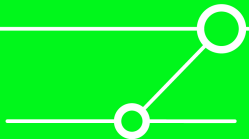
Unterschiedliche Fokussierung der Sicherheitsziele

- Enduser:
Vertraulichkeit (das „emotionale Moment“)
- Ihre Organisation:
Verfügbarkeit, Vertraulichkeit, Datenschutz
- Carrier/Dienst-Anbieter:
Non-Repudiation (Abrechnungs-Betrug!), Lawful Interception,
„Kundenbindung durch Vertrauen“



(Main) Threats

- Abhören von Verbindungen/Sniffing
 - durch Angreifer
 - ‚Lawful‘ (⇔ Anwälte, Journalisten, Beratungsinstitutionen)
- Denial-of-Service
- Kompromittierung von Komponenten
 - => Abhören
 - => Umleiten
- Spoofing
 - => Verlust der Authentizität (Telefon-Banking)
 - => Abrechnungsbetrug



Auf Protokoll-Ebene:

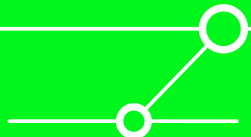
- Implementierungs-Schwächen
- (Hochgradig) Dynamische Kommunikationsbeziehungen
- Fehlende immanente Sicherheitsfeatures

Auf Komponenten-Ebene:

- Unsichere Default-Konfigurationen
- *Design without security in mind*, insbesondere an Kopplungskomponenten
- Seiteneffekte von VoIP Security-Problemen auf ‚Nicht-VoIP Devices‘ (z.B. bei Softphones)

Auf Endgeräte-Ebene:

- Unsichere Default-Konfigurationen
- (Zu) wenig Sicherheits-Features
- Malicious Code ?



Kleine Historie der VoIP-Vulnerabilities

- Vor 18 Monaten:
 - CISCO VoIP-Telefon als Abhöranlage missbraucht
Aufgrund fehlender Sicherheits-Features (etwa fehlender Authentifizierung), schlechter Default-Konfiguration (e.g. aktiviertem Telnet-Zugang) oder mangelhafter Management-Strukturen (Konfig per TFTP etc.), z.B. gegen ältere Cisco 7960-Modelle.
- Vor 12 Monaten:
 - Sniffing = Abhören von VoIP-Telefonaten
Da RTP per default unverschlüsselt überträgt, kann ein Angreifer mit Zugriff auf den Netzwerk-Verkehr (üblicherweise im lokalen Netz per ARP-Spoofing) VoIP-Sitzungen abhören. Das bekannteste Tool ist hier *vomit*.
- aktuell
 - der AOL-Triton – Hack (Buffer Overflow)
ermöglicht die Ausführung von beliebigem Code auf dem „Opfer“-Rechner
- Zukünftig ?

Vor kurzem auf heise.de

ERNW

Wir leben IT-Security.

heise Security - News - Bei Anruf Pufferüberlauf - Mozilla Firefox

File Edit View Go Bookmarks Extras Help

http://www.heise.de/security/news/meldung/75252

Erste Schritte Aktuelle Nachrichten ... Radsport aktiv > FA... ap_sec_ap-client-sec...

heise Security
Sponsored by **Microsoft**

News

Meldung vom 10.07.2006 11:18 [<< Vorige] [Nächste >>]

Bei Anruf Pufferüberlauf

Die quelloffene sipxtapi-Bibliothek von [SIP Foundry](#), die auch in Produkten von [Pingtel](#) sowie in AOLs [AIM Triton](#) eingesetzt wird, verarbeitet bestimmte Felder bei der VoIP-Kommunikation über das Session Initiation Protocol (SIP) nicht korrekt. Angreifer könnten mit manipulierten Clients einen Pufferüberlauf provozieren und sogar Schadcode einschleusen.

Anzeige

fact
Jede Nacht kommen über 20 neue Viren in Umlauf.

fact is
In den SophosLabs geht das Licht niemals aus.

SOPHOS
secured.

Beim Auf- und Abbau von Verbindungen versenden SIP-Clients so

Viren

- WGA-Wurm W32/Cuebot-K
- Trojaner über neue Word-Lücke
- WM-Spielplan enthält Trojaner
- Leap.A für Mac OS X
- E-Mail-Wurm Nyxem

Artikel

- Gefahr aus der Schattenwelt, Teil 2
- Konkurrenz belebt das Geschäft
- Heap-Overflows
- VPN-Knigge
- Schlüssel zum DNS

Tools

- WebScarab

fact
Jede Nacht kommen über 20 neue Viren in Umlauf.

fact is
In den SophosLabs geht das Licht niemals aus.

SOPHOS
secured.

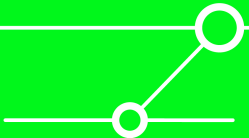
Foren
Fertig

Wie „entsteht“ eine solche Lücke bzw. ihre Publikation?

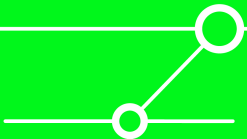
- Auffälliges Verhalten einer Komponente wird beobachtet
 - zufällig (Absturz)
 - durch gezieltes Suchen
 - im Rahmen von Forschungsarbeit
- Eingrenzung des Problems
- Reproduktion / Proof of Concept
- Kontakt zu Hersteller/Autor/Maintainer
- Publikation

Research Techniken

- Fuzzer
- Code Audit
- Reverse Engineering

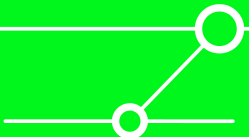


- Fault Injection / Erzeugung von „ungewöhnlichem Input“
- mithilfe von Protocol Fuzzern (SPIKE, Protos)
- oder SIP Testframeworks (SIPp, SipSak, SIPForumTestFramework)
- Kenntnis der Protokolle, File Formate und API notwendig
- Eher für einfache Fehler geeignet

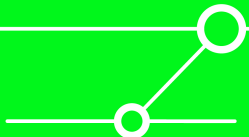


Fuzzer – Vorgehensweise

- Injektion von Fehlern
- Überwachung des Programmverhaltens mit Debugger
- Interessant sind „Programmabstürze“
- Da der injizierte Fehler bekannt ist, kann er auch reproduziert werden (z. B. durch eigenen Code)
- Ggf. Code Audit oder Reverse Engineering

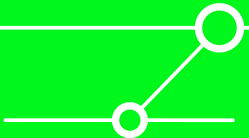


- „Read the source Luke“
- Detaillierte Kenntnis der Programmiersprache notwendig
- Typische Probleme der Sprache, z. B. strcpy() und sprintf() in C
- Komplexe Probleme sind nicht trivial zu entdecken
- Beispiel *ERNW Advisory 02/2006 – SipXtapi Library*



Code Audit

- Tool-Unterstützung möglich:
- RATS
- Splint
- Flawfinder
- Codescan (kommerziell)

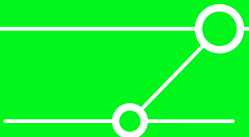


```
1 #define MAXIMUM_INTEGER_STRING_LENGTH 20
2 ...
3 ...
4 ...
5 UtilBoolean SipMessage::getCSeqField(int* sequenceNum, UtilString* sequenceMethod) const
6 {
7     const char* value = getHeaderValue(0, SIP_CSEQ_FIELD);
8     if(value)
9     {
10         // Too slow:
11         /*UtilString sequenceNumString;
12         NameValueTokenizer::getSubField(value, 0,
13             SIP_SUBFIELD_SEPARATORS, &sequenceNumString);
14         *sequenceNum = atoi(sequenceNumString.data());
15
16         NameValueTokenizer::getSubField(value, 1,
17             SIP_SUBFIELD_SEPARATORS, sequenceMethod);*/
18         // Ignore white space in the begining
19         int valueStart = strspn(value, SIP_SUBFIELD_SEPARATORS);
20
21         // Find the end of the sequence number
22         int numStringLength = strcspn(&value[valueStart], SIP_SUBFIELD_SEPARATORS)
23             - valueStart;
24         // Get the method
25         if(sequenceMethod)
26         {
27             *sequenceMethod = &value[numStringLength + valueStart];
28             NameValueTokenizer::frontBackTrim(sequenceMethod, SIP_SUBFIELD_SEPARATORS);
29
30             if(numStringLength > MAXIMUM_INTEGER_STRING_LENGTH)
31             {
32                 osPrintf("WARNING: SipMessage::getCSeqField CSeq number %d characters: %s.\nTruncating to %d\n",
33                     numStringLength, &value[valueStart], MAXIMUM_INTEGER_STRING_LENGTH);
34                 numStringLength = MAXIMUM_INTEGER_STRING_LENGTH;
35             }
36         }
37         if(sequenceNum)
38         {
39             // Convert the sequence number
40             char numBuf[MAXIMUM_INTEGER_STRING_LENGTH + 1];
41             memcpy(numBuf, &value[valueStart], numStringLength);
42             numBuf[numStringLength] = '\0';
43             *sequenceNum = atoi(numBuf);
44         }
45     }
46     else
47     {
48         if(sequenceNum)
49         {
50             *sequenceNum = -1;
51         }
52     }
53 }
```

Wo ist hier das Problem? 😊

Reverse Engineering

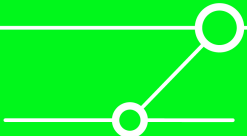
- Binary Audit
- Disassembling
- Prüfen des Assembler Codes
- API Tracing / API Spy
- Aufwendiger Prozess
- Hilfreich: Assembler Know How, OS Architektur und API Know How (Betriebssystemprogrammierung), Prozessor Architektur
- Kenntnisse der eingesetzten Programmiersprache



Reverse Engineering

Auch hier Tool-gestütztes Arbeiten möglich

- API Monitor
- IDA Pro (das wohl wichtigste Tool)
- Bugscam
- BinAudit
- BinNavi
- Bindiff




```
mov ecx, [esp+2Ch+var_1C]
push 14h
push ecx
push ebx ; char
push offset aWarningSipmess ; "WARNING: SipMessage::getCSeqField CSeq "...
call sub_1000C4A0
add esp, 10h
mov ebx, 14h
```

```
loc_1002020F:
mov ecx, [esp+2Ch+arg_4]
test ecx, ecx
jz short loc_1002021E
```

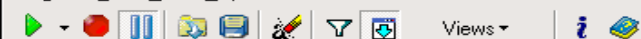
```
loc_100201B7:
mov eax, [esp+2Ch+arg_0]
test eax, eax
jz short loc_1002021E
```

```
push 0
call sub_10009E70
```

```
mov esi, [esp+2Ch+var_1C]
mov ecx, ebx
mov edx, ecx
lea edi, [esp+2Ch+CSeq]
shr ecx, 2
rep movsd
mov ecx, edx
lea eax, [esp+2Ch+CSeq]
and ecx, 3
push eax ; char * - Buffer Overflow in memcpy
rep movsb
mov [esp+ebx+30h+CSeq], 0
call ds:atoi
mov ecx, [esp+30h+arg_0]
add esp, 4
mov [ecx], eax
xor eax, eax
pop edi
pop esi
test ebp, ebp
pop ebp
pop ebx
setnz al
add esp, 1Ch
retn 8
```

```
loc_1002021E:
xor eax, eax
pop edi
test ebp, ebp
pop esi
pop ebp
setnz al
pop ebx
add esp, 1Ch
retn 8
vuln_function endp
```

Noch aufwendiger 😊



Process and Thread ▾ ▾

API Name	Return Value	Module Name	Time Start	IsEntry API
Process and Thread : Process: C:\Daten\Vortraege\LanLine\VoIP\sipXezPhone-0.35a\sipXezPhone.exe(22248) , Thread:21684 (COUNT=6)				
[?] memset	32699768 (0x1F2F578)	MSVCRT.dll	06.07.2006 13:36:47	<input checked="" type="checkbox"/>
[?] memcpy	21048585 (0x1412D09)	MSVCRT.dll	06.07.2006 13:36:48	<input checked="" type="checkbox"/>
[?] memset	32699192 (0x1F2F338)	MSVCRT.dll	06.07.2006 13:36:48	<input checked="" type="checkbox"/>
[?] memcpy	21048622 (0x1412D2E)	MSVCRT.dll	06.07.2006 13:36:48	<input checked="" type="checkbox"/>
[?] memset	32699192 (0x1F2F338)	MSVCRT.dll	06.07.2006 13:36:48	<input checked="" type="checkbox"/>
[?] memcpy	21048656 (0x1412D50)	MSVCRT.dll	06.07.2006 13:36:48	<input checked="" type="checkbox"/>

Summary Information

```

API Name: memcpy
API Define: (Undefine API)
Time Start: 13:36:48.906
Duration: 0,000 ms
Module Name: C:\WINDOWS\system32\MSVCRT.dll
Is Entry API: True
Process: C:\Daten\Vortraege\LanLine\VoIP\sipXezPhone-0.35a\sipXezPhone.exe
Thread: 21684
    
```

Before Call Parameters

```

Pointer Paramter0: 21048656 (0x1412D50)
Pointer Paramter1: 32699192 (0x1F2F338)
Pointer Paramter2: 32 (0x20)
Pointer Paramter3: 5 (0x5)
Pointer Paramter4: 32699760 (0x1F2F570)
Pointer Paramter5: 3 (0x3)
    
```

After Call Parameters

```

Pointer Paramter0: 21048656 (0x1412D50)
Pointer Paramter1: 32699192 (0x1F2F338)
Pointer Paramter2: 32 (0x20)
Pointer Paramter3: 5 (0x5)
Pointer Paramter4: 32699760 (0x1F2F570)
Pointer Paramter5: 3 (0x3)
    
```

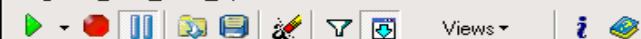
Return

```

21048656 (0x1412D50)
    
```

Call Tree

memcpy



Process and Thread ▾ ▾

API Name	Return Value	Module Name	Time Start	IsEntry API
▶ Process and Thread : Process: C:\Daten\Vortraege\LanLine\VoIP\sipXezPhone-0.35a\sipXezPhone.exe(22248) , Thread: 22280 (COUNT=640)				
▶ Process and Thread : Process: C:\Daten\Vortraege\LanLine\VoIP\sipXezPhone-0.35a\sipXezPhone.exe(22248) , Thread: 22260 (COUNT=3042)				
▼ Process and Thread : Process: C:\Daten\Vortraege\LanLine\VoIP\sipXezPhone-0.35a\sipXezPhone.exe(22248) , Thread: 22020 (COUNT=158)				
LoadLibraryA	2008875008 (0x77BD0000)	kernel32.dll	06.07.2006 13:35:49	<input checked="" type="checkbox"/>
LoadLibraryExA	2008875008 (0x77BD0000)	kernel32.dll	06.07.2006 13:35:49	<input type="checkbox"/>
LoadLibraryExW	2008875008 (0x77BD0000)	kernel32.dll	06.07.2006 13:35:49	<input type="checkbox"/>
LoadLibraryExW	47448065 (0x2D40001)	kernel32.dll	06.07.2006 13:35:52	<input checked="" type="checkbox"/>
LoadLibraryExW	47448065 (0x2D40001)	kernel32.dll	06.07.2006 13:35:52	<input checked="" type="checkbox"/>

Summary Information

API Name: LoadLibraryExW
 API Define: function LoadLibraryExW(lpLibFileName: PWideChar; hFile: THandle; dwFlags: DWORD): HMODULE; stdcall;
 Time Start: 13:35:52.218
 Duration: 0,000 ms
 Module Name: kernel32.dll
 Is Entry API: True
 Process: C:\Daten\Vortraege\LanLine\VoIP\sipXezPhone-0.35a\sipXezPhone.exe
 Thread: 22020

Before Call Parameters

PWideChar lpLibFileName: C:\WINDOWS\system32\calc.exe
 THandle hFile: 0 (0x0)
 DWORD dwFlags: 2 (0x2)

After Call Parameters

PWideChar lpLibFileName: C:\WINDOWS\system32\calc.exe
 THandle hFile: 0 (0x0)
 DWORD dwFlags: 2 (0x2)

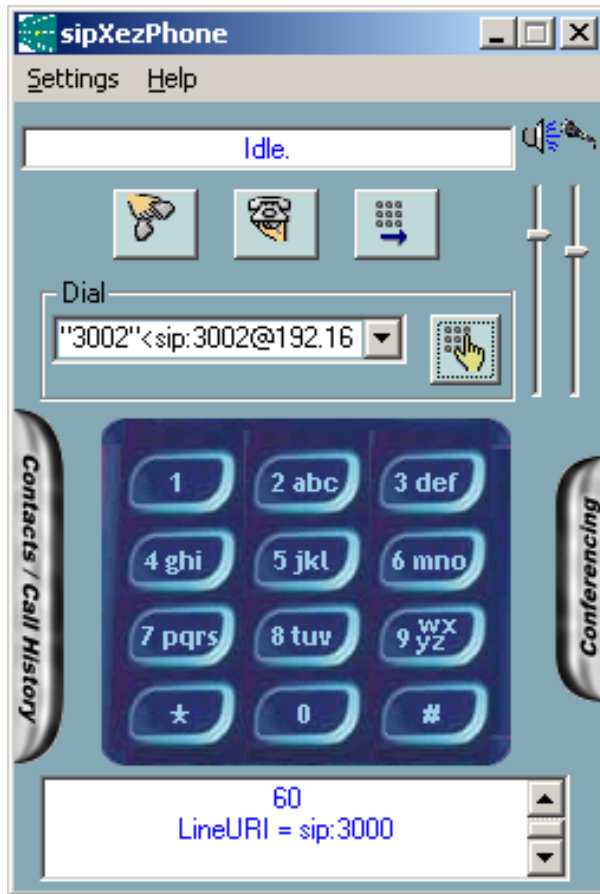
Return

47448065 (0x2D40001)

Call Tree

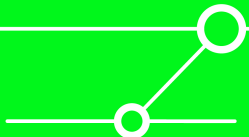
LoadLibraryExW

Buffer Overflow in SIP Foundry's SipXtapi Bibliothek



Kommunikation & Publikation

- Kommunikation an Hersteller
- Übergabe *sehr* detaillierter Infos und Proof of Concept Tools
- Abstimmung des Zeitfensters für die Problembehebung
- Bereitstellung eines Patches / Hotfixes durch den Hersteller
- Und Veröffentlichung ...



Publikation

ERNW

Wir leben IT-Security.

The screenshot shows a Mozilla Firefox browser window with the address bar containing `http://www.securityfocus.com/bid/18906/discuss`. The page title is "SIPfounndry SIPXtapi CSeq Processing Remote Buffer-Overflow Vulnerability". The main content area features a large banner for "Black Hat USA 2006" (July 29-August 3, Las Vegas) and a "Symantec ThreatCon" widget showing a "Level 2: Elevated" threat level. The article title is "SIPfounndry SIPXtapi CSeq Processing Remote Buffer-Overflow Vulnerability". The text describes the vulnerability, stating that SIPXtapi is prone to a remote buffer-overflow vulnerability that can be triggered by a specially crafted 'CSeq' value. It notes that a successful attack could lead to unauthorized remote access. Reports indicate that sipXtapi versions released before March 24, 2006, are vulnerable, including certain PingTel products and AOL Triton versions. A sidebar on the left contains navigation links for "News", "Infocus", "Columnists", "Mailing Lists", "Vulnerabilities", and "Jobs". At the bottom, there is an "ONLINE CLASSIFIEDS" section with a link to "Learn to Simplify Your Network Security".

SecurityFocus™

Black Hat USA 2006
July 29-August 3
Las Vegas
Briefings & Training

Symantec ThreatCon
Level 2: Elevated
Threat level definition

Home | Bugtraq | Vulnerabilities | Mailing Lists | Security Jobs | Tools | Search: [SEARCH]

News
Infocus
Foundations
Microsoft
Unix
IDS
Incidents
Virus
Pen-Test
Firewalls
Columnists
Mailing Lists
Newsletters
Bugtraq
Focus on IDS
Focus on Linux
Focus on Microsoft
Forensics
Pen-test
Security Basics
Vuln Dev
Vulnerabilities
Jobs
Job Opportunities
Resumes
Job Seekers
Employers

SIPfounndry SIPXtapi CSeq Processing Remote Buffer-Overflow Vulnerability

SIPXtapi is reported to be prone to a remote buffer-overflow vulnerability.

This issue presents itself when the application handles a specially crafted 'CSeq' value.

A successful attack may lead to unauthorized remote access in the context of a user running an affected application that uses the vulnerable library.

Reports indicate that sipXtapi versions that were released prior to March 24, 2006 are vulnerable to this issue. Certain PingTel products and versions of AOL Triton may be affected because they employ the vulnerable library.

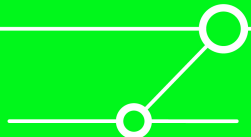
ONLINE CLASSIFIEDS
Learn to Simplify Your Network Security
Download this white paper from Check Point Software Technologies to learn how you can simplify

Master of Science in Information Assurance Online

Fertig

Potentielle Auswirkungen der Schwachstelle

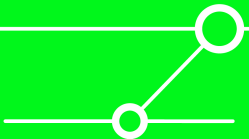
- Anfällig sind (idealerweise: waren) alle Komponenten, die die betroffene Bibliothek nutzen, darunter
 - teilweise Pingtel-Produkte
 - AOL Triton
- Beliebiger Programmcode kann durch Angreifer auf Opfer-System ausgeführt werden (üblicherweise mit den Berechtigungen des Users, der die Komponente gestartet hat)
- Personal Firewall ist nicht zwingend hilfreich (je nach Konfiguration und Kommunikationsverhalten des Users)
- Bei (noch) weiterer Verbreitung von Softphones Potential für Wurm

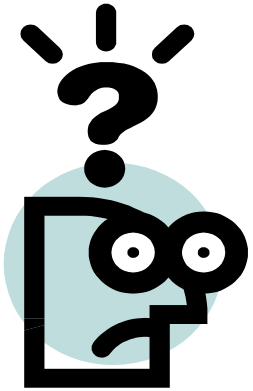


Was lernen wir daraus?



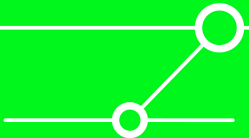
- „VoIP Security“ heisst nicht nur Sicherung des Transports
 - VoIP Security heisst auch gerade Sicherung der Endpunkte sowie Komponenten
 - Hardening, Patchen, Administrations-Prozesse
 - Wie bei den meisten „neuen Technologien“ werden anscheinend grossflächig vorhandene Bibliotheken eingesetzt
- => weitere Sicherheits-Probleme zu erwarten (das ERNW Research Lab arbeitet ;-)





Fragen?

... und Antworten



Vielen Dank für Ihre Aufmerksamkeit!

