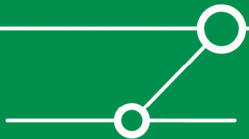


# Hacking VoIP for Fun and Profit

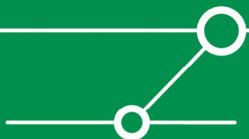
**Michael Thumann & Enno Rey**

**ERNW GmbH**



# Agenda

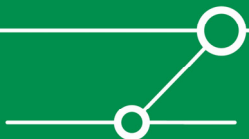
- Vorstellung einer neuen Sicherheitslücke in einer SIP-Bibliothek
- Wie werden solche Lücken entdeckt und publiziert?
- Potentielle Auswirkungen der Schwachstelle
- Was lernen wir daraus?



# ERNW



- Gegründet Sommer 2001 durch Enno Rey
- Netzwerk-Dienstleister mit Sicherheits-Fokus
- Aktuell zehn Mitarbeiter
- Schwerpunkte: Security Management, Audit/Revision, Security Research, Penetrations-Tests
- Kunden: Industrie, Banken, Behörden, Provider



# Gestern auf heise.de...

A screenshot of the heise Security website as it appeared in a Mozilla Firefox browser window. The browser's address bar shows the URL 'http://www.heise.de/security/news/meldung/75252'. The page layout includes a top navigation bar with menu items like 'Datei', 'Bearbeiten', and 'Ansicht'. The main content area features the heise Security logo, a 'Microsoft' sponsorship notice, and a news article titled 'Bei Anruf Pufferüberlauf' dated 10.07.2006. The article text discusses a vulnerability in SIP clients. To the right of the article is a sidebar with sections for 'Viren' (listing threats like WGA-Wurm and Trojaner) and 'Artikel' (listing topics like 'Gefahr aus der Schattenwelt'). A large advertisement for Sophos is prominently displayed in the center and right side of the page, featuring the 'fact' and 'fact is' slogans. The bottom of the browser window shows a 'Foren' section with a 'Fertig' status.

heise Security - News - Bei Anruf Pufferüberlauf - Mozilla Firefox

heise Security  
Sponsored by Microsoft

## News

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

### Bei Anruf Pufferüberlauf

Die quelloffene sipstapi-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

#### Suche

los

#### News

- 7-Tage-Alerts
- 7-Tage-News
- News-Archiv
- Newsletter
- English News

#### Hintergrund

- BSI-Info
- Know-how
- Kommentar
- Praxis
- Produkte
- Hintergrund-Archiv

#### Foren

Fertig

#### 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

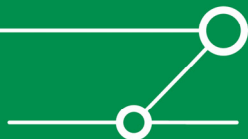
#### Artikel

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

#### Tools

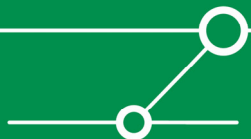
- WebScarab

**SOPHOS**  
secured.



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

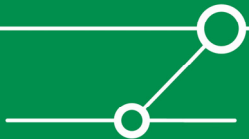
- Auffälliges Verhalten einer Komponente wird beobachtet
  - zufällig (Absturz)
  - durch gezieltes Suchen
  - im Rahmen von Forschungsarbeit
- Eingrenzung des Problems
- Reproduktion / Proof of Concept
- Kontakt zu Hersteller/Autor/Maintainer
- Publikation



# Research Techniken

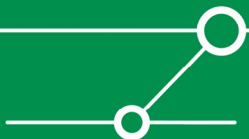


- Fuzzer
- Code Audit
- Reverse Engineering



# Fuzzer

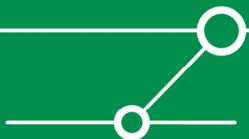
- Fault Injection / Erzeugung von „ungewöhnlichem Input“
- mithilfe von Protocol Fuzzern (SPIKE, Protos)
- oder SIP Testframeworks (SIPp, SipSak, SIPForumTestFramework)
- Kenntnis der Protokolle, File Formate und API notwendig
- Eher für einfache Fehler geeignet



# Fuzzer – Vorgehensweise



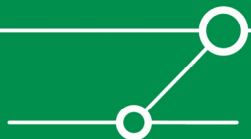
- Injektion von Fehlern
- Überwachung des Programmverhaltens mit Debugger
- Interessant sind „Programmabstürze“
- Da der injizierte Fehler bekannt ist, kann er auch reproduziert werden (z. B. durch eigenen Code)
- Ggf. Code Audit oder Reverse Engineering



# Code Audit



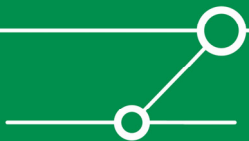
- „Read the source Luke“
- Detaillierte Kenntnis der Programmiersprache notwendig
- Typische Probleme der Sprache, z. B. strcpy() und sprintf() in C
- Komplexe Probleme sind nicht trivial zu entdecken
- Beispiel *ERNW Advisory 02/2006 – SipXtapi Library*



# Code Audit



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



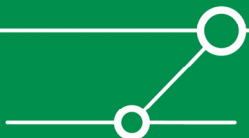
# Code Audit

ERNW

Wir leben IT-Security.

```
0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150
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 = strstr(value, SIP_SUBFIELD_SEPARATORS);
20
21        // Find the end of the sequence number
22        int numStringLength = strstr(&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 }
```

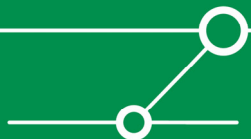
Wo ist hier das Problem? 😊



# Reverse Engineering



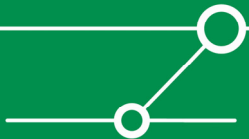
- Binary Audit
- Disassembling
- Prüfen des Assembler Codes
- API Tracing / API Spy
- Aufwendiger Prozess
- Hilfreich: Assembler Know How, OS Architektur und API Know How (Betriebssystemprogrammierung), Prozessor Architektur
- Kenntnisse der eingesetzten Programmiersprache



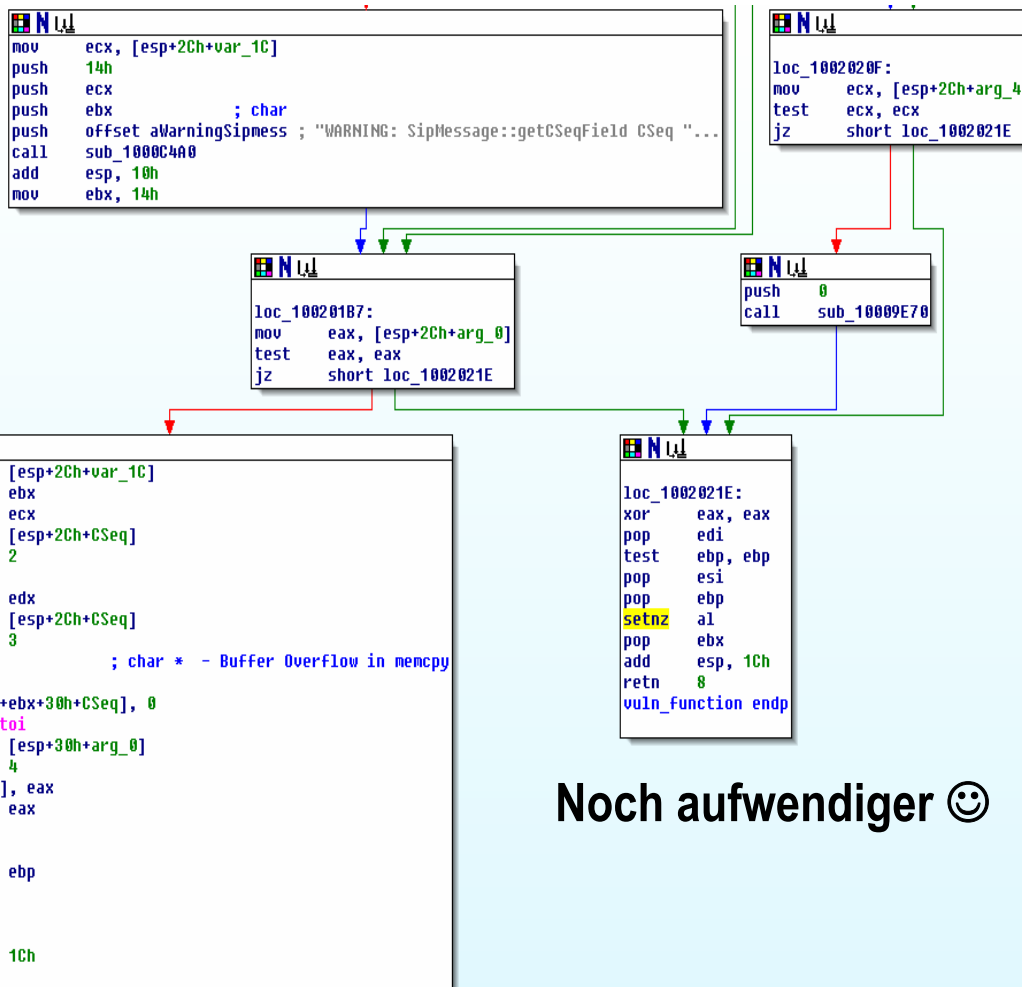
# Reverse Engineering

Auch hier Tool-gestütztes Arbeiten möglich

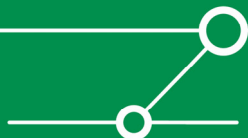
- API Monitor
- IDA Pro (das wohl wichtigste Tool)
- Bugscam
- BinAudit
- BinNavi
- Bindiff



# Reverse Engineering



Noch aufwendiger 😊



# Reverse Engineering

ERNW

Wir leben IT-Security.

The screenshot shows the API Monitor application window. The top menu bar includes Action, File, Grid, and Help. Below the menu is a toolbar with various icons. The main window is divided into several sections:

- Process and Thread:** A dropdown menu showing the current process and thread.
- API List:** A table with columns for API Name, Return Value, Module Name, Time Start, and IsEntry API. The table contains several entries for memset and memcpy.
- Summary Information:** A detailed view of the selected API call (memcpy). It includes fields for API Name, API Define, Time Start, Duration, Module Name, Is Entry API, Process, and Thread.
- Before Call Parameters:** A list of parameters passed to the function before the call.
- After Call Parameters:** A list of parameters passed to the function after the call.
- Return:** The return value of the function.
- Call Tree:** A tree view showing the call stack.

API Name	Return Value	Module Name	Time Start	IsEntry API
(?) 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 Parameter0: 21048656 (0x1412D50)
- Pointer Parameter1: 32699192 (0x1F2F338)
- Pointer Parameter2: 32 (0x20)
- Pointer Parameter3: 5 (0x5)
- Pointer Parameter4: 32699760 (0x1F2F570)
- Pointer Parameter5: 3 (0x3)

**After Call Parameters**

- Pointer Parameter0: 21048656 (0x1412D50)
- Pointer Parameter1: 32699192 (0x1F2F338)
- Pointer Parameter2: 32 (0x20)
- Pointer Parameter3: 5 (0x5)
- Pointer Parameter4: 32699760 (0x1F2F570)
- Pointer Parameter5: 3 (0x3)

**Return**

- 21048656 (0x1412D50)

**Call Tree**

- memcpy

# Reverse Engineering

ERNW

Wir leben IT-Security.

The screenshot shows the API Monitor application window. The main pane displays a list of API calls with columns for API Name, Return Value, Module Name, Time Start, and IsEntry API. The selected entry is LoadLibraryExW, which is detailed in the Summary Information pane below.

API Name	Return Value	Module Name	Time Start	IsEntry API
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)

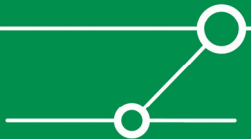
**Return**

- 47448065 (0x2D40001)

**Call Tree**

- LoadLibraryExW

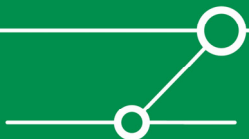
# Demo



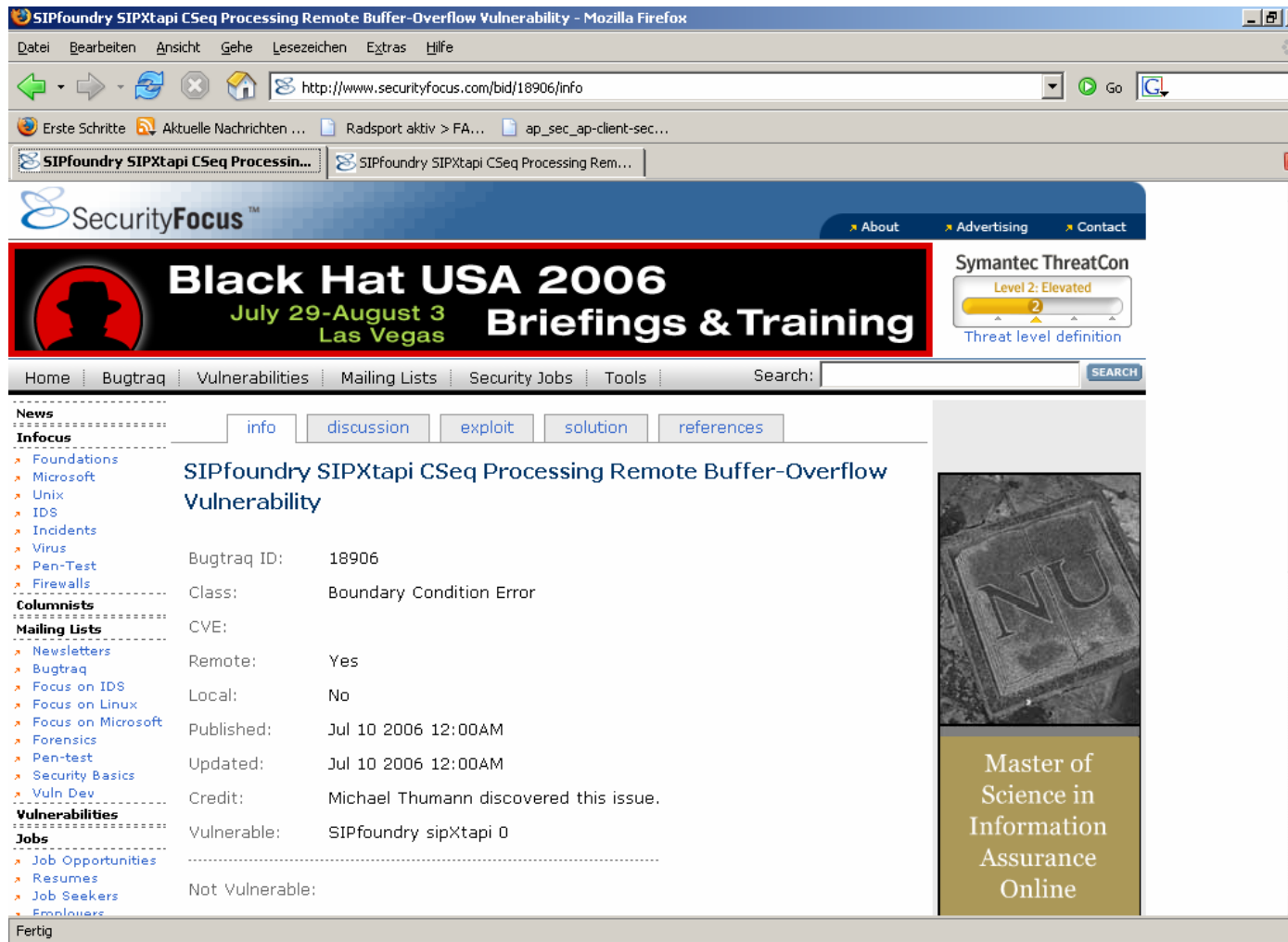
# Kommunikation & Publikation



- 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 ...



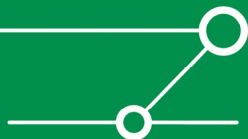
# Publikation



The screenshot shows a Mozilla Firefox browser window with the address bar containing `http://www.securityfocus.com/bid/18906/info`. The page title is "SIPfoundry SIPXtapi CSeq Processing Remote Buffer-Overflow Vulnerability". The main content area displays the following details:

Bugtraq ID:	18906
Class:	Boundary Condition Error
CVE:	
Remote:	Yes
Local:	No
Published:	Jul 10 2006 12:00AM
Updated:	Jul 10 2006 12:00AM
Credit:	Michael Thumann discovered this issue.
Vulnerable:	SIPfoundry sipXtapi 0
Not Vulnerable:	

The article title is "SIPfoundry SIPXtapi CSeq Processing Remote Buffer-Overflow Vulnerability". The page also features a "Symantec ThreatCon" widget showing a "Level 2: Elevated" threat level and a "Black Hat USA 2006" banner for July 29-August 3 in Las Vegas.



# Publikation

SIPfounndry SIPXtapi CSeq Processing Remote Buffer-Overflow Vulnerability - Mozilla Firefox

http://www.securityfocus.com/bid/18906/discuss

## 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.

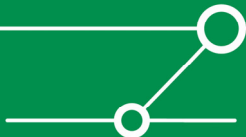
Reports indicate that sipXtapi versions that were released prior to March 24, 2006 are vulnerable to this issue. Certain PingTel products and versions of AOL Triton may be affected because they employ the vulnerable library.

**ONLINE CLASSIFIEDS**

[Learn to Simplify Your Network Security](#)  
Download this white paper from Check Point Software Technologies to learn how you can simplify

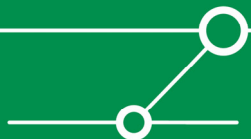
Master of Science in Information Assurance Online

Fertig



# Potentielle Auswirkungen der Schwachstelle

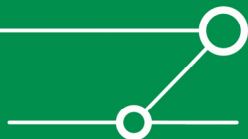
- Anfällig sind (idealerweise: waren) alle Komponenten, die die betroffene Bibliothek nutzen, darunter
  - teilweise Pingtel-Produkte
  - AOL Triton
- Beliebiger Programmcode kann durch Angreifer auf Opfer-System ausgeführt werden (üblicherweise mit den Berechtigungen des Users, der die Komponente gestartet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

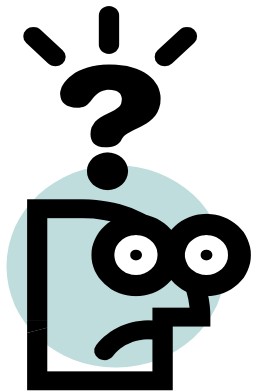


## Was lernen wir daraus?

- „VoIP Security“ heisst nicht nur Sicherung des Transports
- ... für die sich aber scheinbar sowieso niemand interessiert.  
[Diplomarbeit bei ERNW brachte interessante Erkenntnisse hinsichtlich SRTP-Interoperabilität]
- VoIP Security heisst auch+gerade Sicherung der Endpunkte sowie Komponenten
- Hardening, Patchen, Administrations-Prozesse
  
- Wie bei den meisten „neuen Technologien“ werden scheinbar grossflächig vorhandene Bibliotheken eingesetzt (remember OpenSSL?)

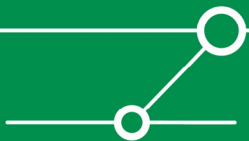
=> weitere Sicherheits-Probleme zu erwarten (das ERNW Research Lab arbeitet ;-)





Fragen?

... und Antworten



**Vielen Dank für Ihre Aufmerksamkeit!**

