Global Information Assurance Certification Paper

Building an Enterprise Ready, Client based VPN Solution.
And doing it on the cheap.

By Kurt Anderson

06/08/04

Introduction

As the proliferation of the internet and the use of computers continues to expand in both the corporate world and private sector so grows the need for secure communication. Secure communication comes in many forms, encrypted e-mail, SSL secured web sessions to name a few, all of which carry their own pros and cons. Most of these methods of security are aimed at protecting transmission over very specific forms of communication like e-mail, web, and telnet.

Large business have become ever more reliant on computers and connecting their global and regional offices to help cut costs and share data over a broad range of departments and affiliates. Smaller business use remote access to keep their competitive edge. In these settings the VPN (Virtual Private Networks) has proved invaluable as a method of connecting remote sites and users back to their home networks and doing so in a secure manner. This paper will give you a brief overview on remote access and the VPN and show you how to build your own VPN solution and do it with free software and hardware you most likely already have.

A little history on Remote access and the VPN

Remote access over the years has been a slowly evolving entity. Access to home, small office and corporate networks have, since the start of "remote computing" been primarily been over POTS (Plain Old Telephone Service) on a dedicated line connected to a individual server or into a RAS (Remote access server) allowing access to a broader range of network resources. While being consistently reliable and certainly secure it tied up expensive dial-in lines, required expensive hardware and was painfully slow. This all changed with the advent VPN. VPNs use high-encryption over increasingly common high speed, always on internet connections (broadband at home and in the small office and T1/T3s at the corporate level) to create a secure tunnel over unsecured communication channels. The best part is that VPNs use the existing infrastructure (the internet) as the medium to connect over and can show a significant savings compared to RAS over POTS.

The VPN (Virtual private network) has what I believe was had a profound impact on what remote access was and is still shaping the remote access solutions of tomorrow.

Why I chose PPTP over IPsec

There are a number of VPN technologies in use today, with the two most prominent probably being PPTP (Point to Point Tunneling Protocol) and IPsec. Both are open standards developed by the IETF (Internet Engineering Task

Force)[1] consisting of a consortium of key vendors and individuals in the computer industry.

IPsec is an application layer based security technology known for its reliability and strong encryption. The big downside of IPsec, in some regards, is its requirement of external software to work as IPsec works at the application layer. Free IPsec implementations do exist but for this reason IPsec has found a home more in protecting server to server connections than fulfilling remote access needs of individual users. This is where PPTP comes in.

PPTP is the VPN solution (or technology) for all the road warriors, vendors and stay-at-home employees out there. PPTP can support server to server connections as well but its design was really for temporary connection types. Unlike IPsec which is application based PPTP is a protocol and resides at the network layer level. PPTP acts as a VPN by encapsulating PPP (Point to Point) protocol and tunneling it through a given IP network. All communication including authentication and encryption are handled by PPP.[2] What that means is you can build support for PPTP connections right into the OS. And that is exactly what Microsoft did when it created its own "version" of PPTP and included a PPTP client (by default or download) in all of its Windows products from Windows 95 to Windows XP. For this reason alone PPTP makes a great client-server VPN solution, in that MS Windows owns the lions share of the desktop computer OS market. With client software already installed on the remote end all that is left is a PPTP server on the network side. While Microsoft does include a PPTP server in its server offerings starting with Windows NT Server, we're going to show you how you can leverage a FREE PPTP server to reduce your remote access costs even further.

This is where the Linux PoPToP Project comes in.

The Linux PoPToP Project

Building a free PPTP server that can support MS Windows PPTP clients natively is where PoPToP comes in. Poptop was originally developed by Morton Bay, now CyberGuard () and released to the public under the GPL (General Public License) in 1999. The project is now located at http://www.poptop.org and those involved have created a scaleable PPTP server solution capable of running on Linux, Solaris, FreeBSD and OpenBSD. The implementation of PPTP developed through the PoPToP Project works seamlessly with Microsoft's implementation of PPTP[3] that is included in most all of its windows products. As you can see already having a client installed on the remote computers and using an open source PPTP server running on Linux leaves only for hardware to be

---

[1] The full RFCs can be found at http://www.ietf.org/rfc.html. Put the RFC # associated with either PPTP or IPsec in the search field or go to the index and scroll to find them.

[2] Ramsay, Matthew. "Poptop, a Secure and Free VPN Solution." Linux Journal June 2000

[3] Microsoft. "Point-Point Tunneling Protocol FAQ" June 27 2001. URL: http://www.microsoft.com/ntserver/ProductInfo/faqs/PPTPfaq.asp

acquired to complete the entire solution. Using a stripped down version of Linux like we will install in just a little on that old PIII 450 lying in the corner of your office should allow for 100-200 simultaneous connections into your office environment. The only real limitation ends up being bandwidth on the client end and that of your corporate network or small office.

Things to note before the install

The first and foremost is that renowned cryptologist Bruce Schneier has shown that while in the PPTP protocol itself they have found no flaws, serious flaws do exist in Microsoft's initial implementation of it. In fact he states, "There are not one but SIX serious flaws".[4] Since the drive of the PoPToP Project was to create a solution that allows Microsoft Windows clients to connect to a Linux PPTP server it too has those same flaws. Thankfully there are a number of things we can do to mitigate the risk of these flaws being exploited. Microsoft's version of CHAP (Challenge Handshake Authentication Protocol) called MSCHAP has been reworked to correct the flaws in it and released again now as MSCHAPv2. MPPE (Microsoft's Point to Point Encryption) was reworked as well to Schneier's satisfaction. We will also include in the installation how to specify that clients use these new features and don't fall back on the old, flawed authentication and encryption protocols.

The installation

Preliminary Preparation and Requirements

This paper assumes that RH (Red Hat 9) has been loaded on a machine meeting the minimum hardware standards above and the additional packages specified under "Kernel Development" were either included in the initial install of RH or added after the install was complete.

Note: The "Kernel Development" packages are included for possible recompilation of the kernel in the future.

In addition to the above the following packages should be obtained at the sites listed below. You can download them to whatever directory you would normally store source code in, for this paper I created a directory in /root called "pptpsrc" and downloaded the files there.

Kernelmod-0.7.1.tar.gz

http://prdownloads.sourceforge.net/poptop/kernelmod-0.7.1.tar.gz?download

Note: This installs MPPE which is Microsoft's Point-to-Point Encryption. Since we are planning on using Microsoft's PPTP client and the client uses MPPE we need to add it to our server install.

pptpd-1.1.3-20030409.tar.gz

[4] Schneier, Bruce, "Frequently asked Questions -- Microsoft's PPTP Implementation" 1998 URL: http://www.schneier.com/pptp-faq.html

http://prdownloads.sourceforge.net/poptop/pptpd-1.1.3-20030409.tar.gz?download
Note: This is the actual PPTP server source from the Poptop project.

Hardware requirements
Although the PoPToP Project does not specify any real minimum hardware specs there are a few recommendations. For a 10-50 user (simultaneous connections) environment a PII 200 with 256 of RAM should provide more than enough power for a solid VPN server. If you're expecting to have as many as 250-350 users a PIII 750 with 512MB - 1GB of RAM should be utilized.[5]

Note: Your mileage will vary. Due to unique environmental variables in each separate network the machine you end up using might not provide the necessary processing power needed to keep the clients connection robust or alive.

The installation is comprised of (6) steps

INSTALL NOTE: You must be logged in as the root user for the installation to be successful. When asked to enter a command it will always be found in bold and after the "#".

Step 1 – Install the Kernelmod

```
[root@moneypenny2 pptpsrc]# gunzip kernelmod-0.7.1.tar.gz
[root@moneypenny2 pptpsrc]# tar -xvf kernelmod-0.7.1.tar
[root@moneypenny2 pptpsrc]# cd kernelmod
[root@moneypenny2 kernelmod]# ./kernelmod.sh
```

Verify the kernelmod installed correctly by entering the following

```
[root@moneypenny2 kernelmod]# modprobe ppp-compress-18

Warning: loading /lib/modules/2.4.20-
8/kernel/drivers/net/ppp_mppe.o will taint the kernel: non-
GPL license - BSD without advertisement clause
  See http://www.tux.org/lkml/#export-tainted for
information about tainted modules
Module ppp_mppe loaded, with warnings
```

INSTALL NOTE: The above message is normal and expected. The last line confirms that the module did load. You can further confirm that it has in fact loaded by entering the following and look "ppp_mppe" and "ppp_generic" (in italics) under the module heading.

```
[root@moneypenny2 kernelmod]# lsmod
Module                  Size  Used by    Tainted: P
ppp_mppe               13720   0  (unused)
```

[5] http://sourceforge.net/mailarchive/message.php?msg_id=6860884

```
ppp_generic             23836   0  [ppp_mppe]
slhc                     6580   0  [ppp_generic]
autofs                  12148   0  (autoclean) (unused)
tlan                    29528   1
keybdev                  2720   0  (unused)
mousedev                 5204   0  (unused)
hid                     20772   0  (unused)
input                    5632   0  [keybdev mousedev hid]
usb-uhci                24652   0  (unused)
usbcore                 73088   1  [hid usb-uhci]
ext3                    64704   2
jbd                     47828   2  [ext3]
[root@moneypenny2 kernelmod]#
```

Step 2 – Install the PPTP server

```
[root@moneypenny2 kernelmod]# cd ..
[root@moneypenny2 pptpsrc]# gunzip pptpd-1.1.4-b4.tar.gz
[root@moneypenny2 pptpsrc]# cd poptop-1.1.4/
[root@moneypenny2 poptop-1.1.4]# f
```

INSTALL NOTE: Make sure that the there are no errors before continuing. If there are errors, resolve those first and then run configure again.

```
[root@moneypenny2 poptop-1.1.4]# make
[root@moneypenny2 poptop-1.1.4]# make install
```

INSTALL NOTE: Make sure that the there are no errors before continuing. If there are errors, resolve those first and then run configure again.

Step 3 – Configure the PPTP server parameters and start the service

INSTALL NOTE: These parameters should work for Windows 2000 & XP clients without modifications. If you have other clients in your environment you might have to modify the parameter files slightly. You can find documented examples in the PoPToP source folder under /samples.

Edit the file pptpd.conf located in /etc include the following:

```
option /etc/ppp/options.pptpd
localip xxx.xx.xxx.xxx
remoteip xxx.xx.xxx.xxx
```

INSTALL NOTE: The “xxx.xx.xxx.xxx” is the IP address or range you will use for your local server and for your remote clients. If you have dedicated a whole subnet to your remote users use a “-“ to encompass the whole subnet (Example: 172.23.129.1-254).

Create the file options.pptpd in /etc/ppp to include the following:

```
[name *]
[lock]
[mtu 1450]
[mru 1450]
[proxyarp]
[auth]
[+chap]
#[+chapms]
[+chapms-v2]
[ipcp-accept-local]
[ipcp-accept-remote]
[lcp-echo-failure 3]
[lcp-echo-interval 5]
[deflate 0]
[mppe-128]
#[mppe-40]
[mppe-stateless]
```

INSTALL NOTE: Notice that we have specifically commented out the original implementation of MSCHAP and MPPE 40-Bit encryption. We are forcing our clients to authenticate using CHAP (Not MS) or MSCHAPv2 and only allow 128-Bit encryption. This reasonably mitigates the problems we discussed earlier with Microsoft's "flavor" of PPTP.

For a detailed description of all the options used above, refer to the sample options file in the pptpsrc/poptop/samples folder.

Finally we need to edit the file /etc/ppp/chap-secrets which includes the userid and passwords we will give to our remote users. In this instance I've added the below.

```
# Secrets for authentication using CHAP
# client(user) server(server)  secret(pass) IP(if fixed)
john           moneypenny2     dial1254wu
```

Lastly it's time to start the service by issuing the command below

```
[root@moneypenny2 ppp]# pptpd
```

Step 4 – Configure the Firewall to allow PPTP traffic to pass to the server

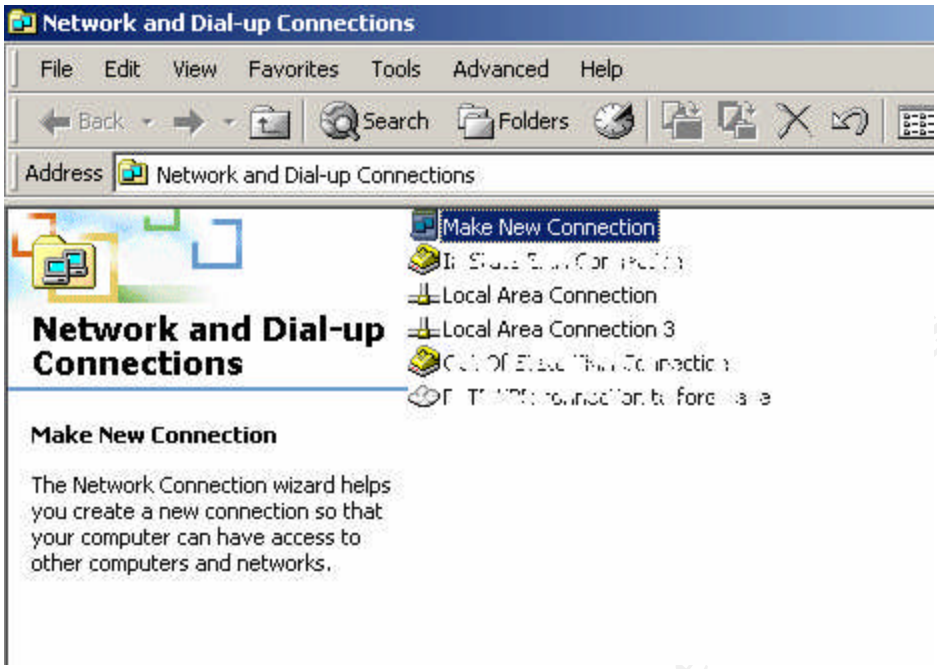
Since the PPTP server will reside inside the firewall you will need to create a rule on your firewall or router forwarding all traffic for port 1723 to your PPTP server. Some firewalls with the label "SOHO" (Small Office Home Office) might have a rule already written labeled "PPTP" that only needs to be modified with the server name and then turned on.

However, ultimately how this is facilitated depends on what type of firewall or router you utilize and is outside the scope of this paper.

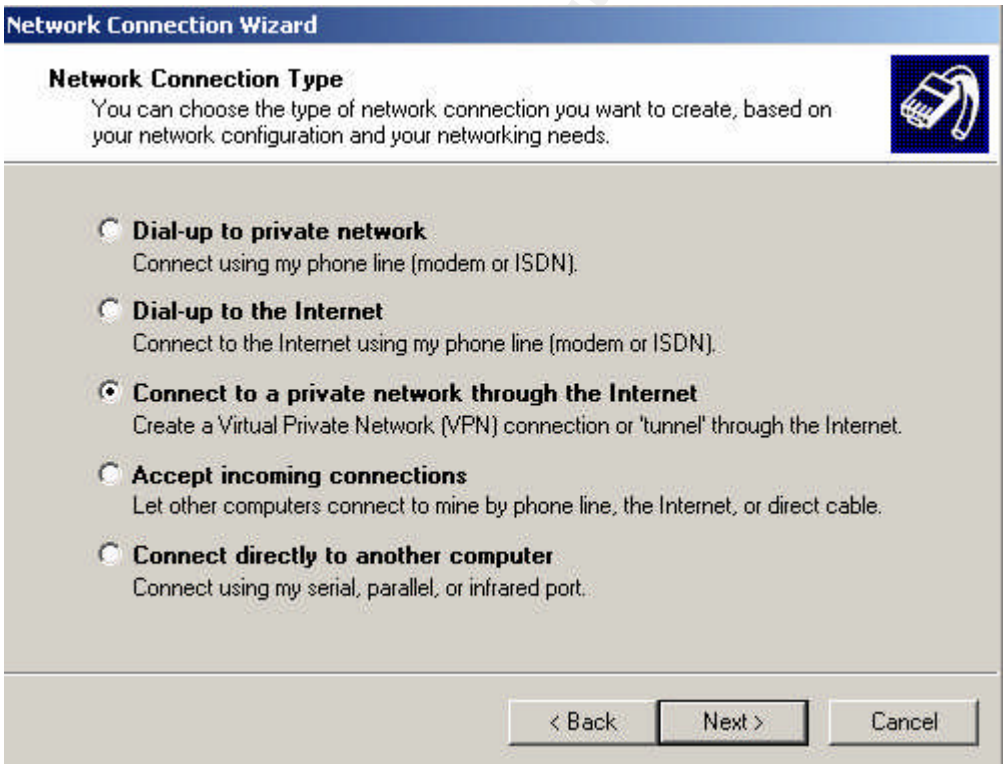
Step 5 – Configure the PPTP Client (In this example the native Windows 2000 PPTP Client)

On the Windows desktop right click on “My Network Places” and select “Properties” (see below)

Once the “Network and Dial-up Connections” window opens select “Make New Connection” (see below)



The new connection wizard will open and you'll want to select "Connect to a private network through the Internet" (see below)

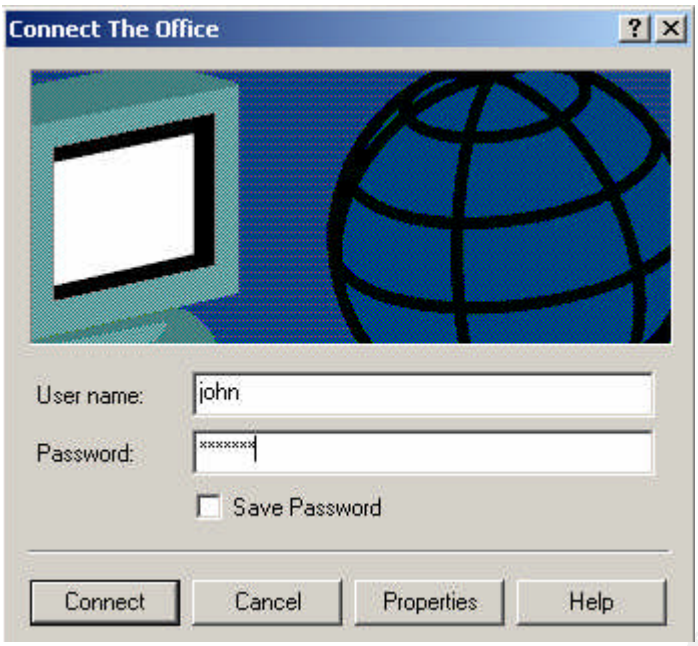


Follow the prompts entering the information for your connection which includes the IP address of your firewall (which will route your VPN traffic to the server for you) and the name of your connection (in this instance "The Office"). At the end

it will ask you to click “o.k.” and will place a shortcut on your desktop for your connection. (See below)

Step 6 – Connect

Our client now away from the office and connected to some form of high speed internet has only to click the shortcut created, enter his or her credential information (as seen below) and connect. The user is authenticated via our PPTP server and allowed seamlessly into the network.



Conclusion:

The PPTP Server that came out of the PoPToP Project allows business and home users alike to build an affordable, scaleable and secure remote access solution using free software and inexpensive hardware you probably already have. This solution should give everyone pause to reevaluate their remote access needs, current installations and the places they wish to go tomorrow.

Glossary of Terms:

PPP: Point-to-Point Protocol

PPTP: Point-To-Point Tunneling Protocol

POTS: Plain Old Telephone Service

VPN: Virtual Private Network

IETF: Internet Engineering Task Force - http://www.ietf.org/

WAN: Wide Area Network

GPL: General Public License - http://www.gnu.org/copyleft/gpl.html

RH: Red Hat

MPPE: Microsoft's Point-to-Point Encryption

CHAP: Challenge Handshake Authentication Protocol

MSCHAP: Microsoft CHAP

RAS: Remote Access Server or Remote Access Service

Sources:

Schneier, Bruce. “Frequently asked Questions -- Microsoft’s PPTP Implementation” 1998 URL: http://www.schneier.com/pptp-faq.html (10 May, 2004)

Microsoft. “Point-Point Tunneling Protocol FAQ” June 27 2001. URL: http://www.microsoft.com/ntserver/ProductInfo/faqs/PPTPfaq.asp (May 10, 2004)

Poptop Project. “Setting up PPTP on Linux Kernel 2.4 HOWTO” Version 0.76. December 16, 2001. URL: http://poptop.sourceforge.net/dox/source-howto.html (May 6, 2004)

Poptop Project. “Poptop Questions & Answers” June 30, 2003. URL: http://poptop.sourceforge.net/dox/qna.html (May 6, 2004)

Poptop Project Mailing list archive. URL: http://sourceforge.net/mailarchive/forum.php?forum_id=8250 (March 2002 – Present day)

Ramsay, Matthew. “Poptop, a Secure and Free VPN Solution.” Linux Journal June 2000 (2000)

Cameron, James. “Red Hat 9 HOWTO” June 18, 2003. URL: http://pptpclient.sourceforge.net/howto-redhat-90.phtml (May 7, 2004)

McCabe, Linus & Cameron, James. “Routing HOWTO” November 25, 2002. URL: http://pptpclient.sourceforge.net/routing.phtml (May 7, 2004)

IETF. “Point-to-Point Tunneling Protocol (PPTP)” July 1999. URL: http://www.ietf.org/rfc/rfc2637.txt (May 10, 2004)

Last Updated: June 24th, 2018

Upcoming Training

SANS Cyber Defence Canberra 2018	Canberra, Australia	Jun 25, 2018 - Jul 07, 2018	Live Event
SANS Vancouver 2018	Vancouver, BC	Jun 25, 2018 - Jun 30, 2018	Live Event
SANS Minneapolis 2018	Minneapolis, MN	Jun 25, 2018 - Jun 30, 2018	Live Event
Community SANS Nashville SEC401	Nashville, TN	Jun 25, 2018 - Jun 30, 2018	Community SANS
SANS London July 2018	London, United Kingdom	Jul 02, 2018 - Jul 07, 2018	Live Event
SANS Cyber Defence Singapore 2018	Singapore, Singapore	Jul 09, 2018 - Jul 14, 2018	Live Event
SANSFIRE 2018	Washington, DC	Jul 14, 2018 - Jul 21, 2018	Live Event
SANSFIRE 2018 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 16, 2018 - Jul 21, 2018	vLive
Mentor Session - SEC401	Jacksonville, FL	Jul 17, 2018 - Aug 28, 2018	Mentor
Community SANS Annapolis Junction SEC401	Annapolis Junction, MD	Jul 23, 2018 - Jul 28, 2018	Community SANS
SANS Riyadh July 2018	Riyadh, Kingdom Of Saudi Arabia	Jul 28, 2018 - Aug 02, 2018	Live Event
SANS Pittsburgh 2018	Pittsburgh, PA	Jul 30, 2018 - Aug 04, 2018	Live Event
SANS Boston Summer 2018	Boston, MA	Aug 06, 2018 - Aug 11, 2018	Live Event
SANS August Sydney 2018	Sydney, Australia	Aug 06, 2018 - Aug 25, 2018	Live Event
San Antonio 2018 - SEC401: Security Essentials Bootcamp Style	San Antonio, TX	Aug 06, 2018 - Aug 11, 2018	vLive
SANS San Antonio 2018	San Antonio, TX	Aug 06, 2018 - Aug 11, 2018	Live Event
SANS Hyderabad 2018	Hyderabad, India	Aug 06, 2018 - Aug 11, 2018	Live Event
Mentor Session - SEC401	Ankara, Turkey	Aug 08, 2018 - Oct 03, 2018	Mentor
Northern Virginia- Alexandria 2018 - SEC401: Security Essentials Bootcamp Style	Alexandria, VA	Aug 13, 2018 - Aug 18, 2018	vLive
SANS Northern Virginia- Alexandria 2018	Alexandria, VA	Aug 13, 2018 - Aug 18, 2018	Live Event
SANS New York City Summer 2018	New York City, NY	Aug 13, 2018 - Aug 18, 2018	Live Event
SANS Virginia Beach 2018	Virginia Beach, VA	Aug 20, 2018 - Aug 31, 2018	Live Event
SANS Chicago 2018	Chicago, IL	Aug 20, 2018 - Aug 25, 2018	Live Event
Mentor Session AW - SEC401	Raleigh, NC	Aug 22, 2018 - Aug 29, 2018	Mentor
SANS San Francisco Summer 2018	San Francisco, CA	Aug 26, 2018 - Aug 31, 2018	Live Event
SANS Tokyo Autumn 2018	Tokyo, Japan	Sep 03, 2018 - Sep 15, 2018	Live Event
SANS Amsterdam September 2018	Amsterdam, Netherlands	Sep 03, 2018 - Sep 08, 2018	Live Event
SANS Tampa-Clearwater 2018	Tampa, FL	Sep 04, 2018 - Sep 09, 2018	Live Event
SANS Baltimore Fall 2018	Baltimore, MD	Sep 08, 2018 - Sep 15, 2018	Live Event
SANS vLive - SEC401: Security Essentials Bootcamp Style	SEC401 - 201809,	Sep 11, 2018 - Oct 18, 2018	vLive
SANS Munich September 2018	Munich, Germany	Sep 16, 2018 - Sep 22, 2018	Live Event