



Training Module 2 - Keyboard-KVM Vulnerabilities

Version B
July 27, 2011

Targeted attacks on KVM?

What is a targeted attack?

Some IT attacks are random. The attacker scans large number of unknown computers to find specific vulnerabilities and then use that vulnerability to attack one or more computers. A targeted attack is an attacker or a group of attackers that specifically targeting your organization. They may spend months or even years in targeted and structured attempts to gain access to your networks.

Why targeted attack using a KVM?

Just because they are there – one leg in a classified network and another leg in non-classified network. It is just bad timing and location...

What is the target on the attack using a KVMs?

Most users think that the goal of such attacker is to capture the information that the user types. This is definitely not the case today! Most targeted attacks using KVMs are intended to create a bridge or leak between classified isolated network and non-classified network.

How would I know that my KVMs are being attacked?

You will not know.

Targeted attacks on KVM?

What else can be attacked to leak?

Nothing. Just the KVMs. There is no other equipment with the potential to leak. Attackers would never target a network diode or a pump as they know that it would be an impossible task.

Where is the attacker located?

Typically in a remote site (different country, different continent). The magic of the internet...

Who is the potential attacker?

Depending on the value of your classified information. It can range from one or two team members to large number of specialized experts working in an agency in another country. It is definitely not a bored student in California...

How long does it take?

Anything from weeks to years depending on how important is the information and how vulnerable your organization.

Targeted attacks on KVM?

Is the attack is KVM model specific?

There are many similarities and common vulnerabilities across different product. Still when secure KVMs involved the attack must be very much model dependant.

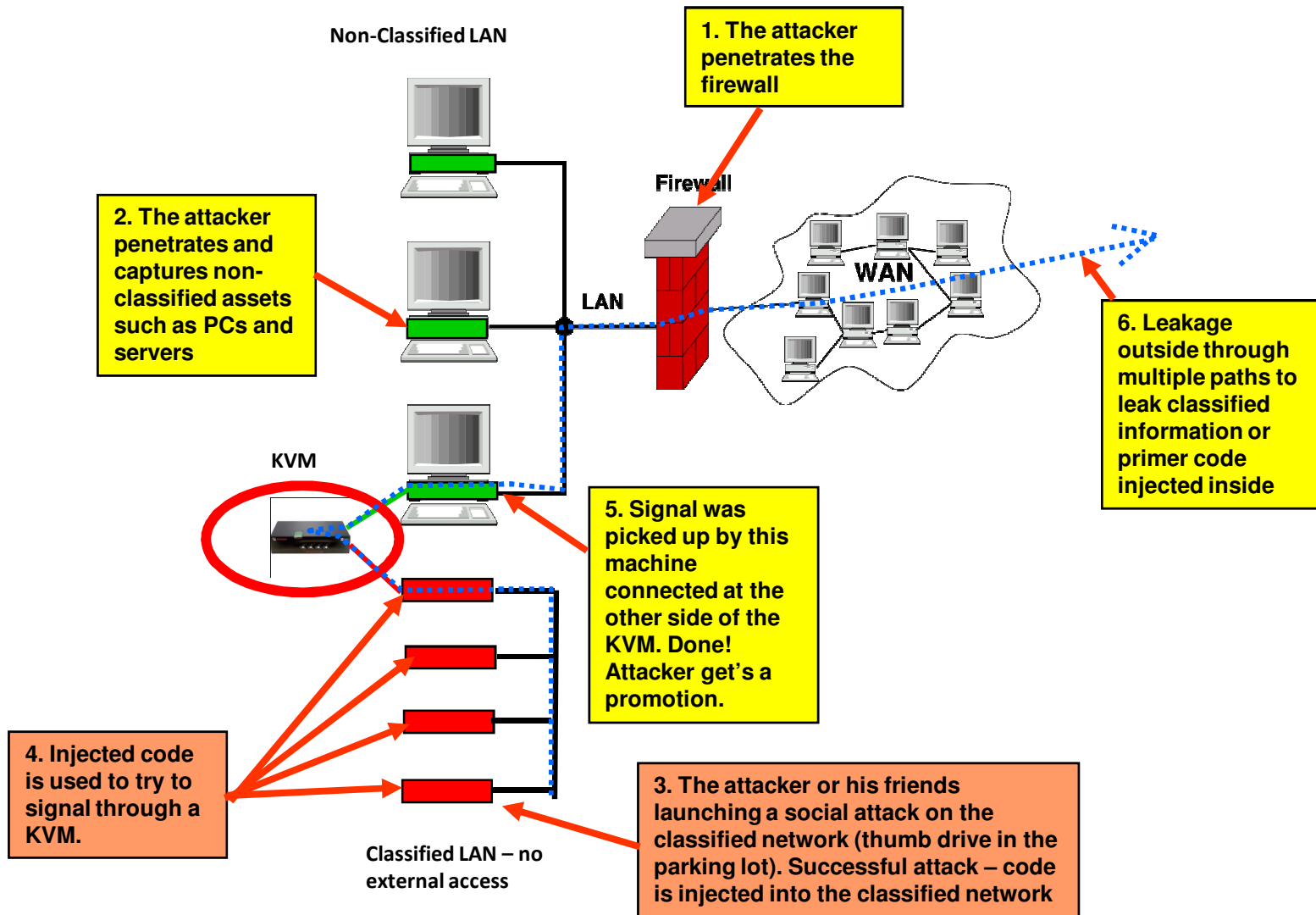
How would the attacker know what KVM models I have?

Sometimes he / she can find that information on the internet (bids, solicitations). Sometime he / she will guess and try. In most cases the attacker will run an agent that will detect the footprint of the KVM and report back the findings.

So what is the concept?

1. Typically the attacker gains access to the non-classified network that is connected to the internet. This is done through firewall penetration. Not too complex for professional attackers.
2. Next step is to inject some code into the classified network. As this network is isolated – the attacker typically use social attack. The code that injected is used to provide signaling to the other side of the KVM.
3. The attacker scans the non-classified network with an attempt to receive the signaling.
4. Once signaling is received, the attacker knows that that KVM is located in the critical position. The leakage starts from that point.

KVM Remote attack and leakage scenario



Targeted attacks on KVM?

Can I use IDS or anti-virus software to detect this attack?

No. The malicious code involved in these attacks are not well detected and profiled. This is a custom code made by professionals with very specific intentions. They know all about your IDS.

How the KVM user is involved in such attack?

Typically the user is not involved at all. The attack is abusing his / her equipment. Nothing else. In most cases the user is not aware of the attack or the resulted leak.

What can I do to protect from such attack?

1. Use latest high security KVMs only. Remove everything else.
2. Segment your networks.
3. Protect your organization from social attacks by proper training and tools.

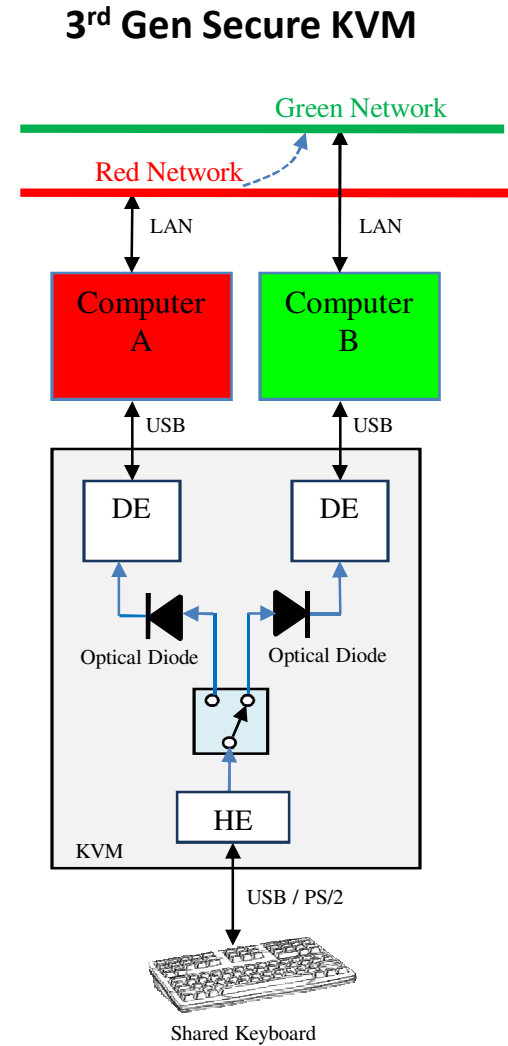
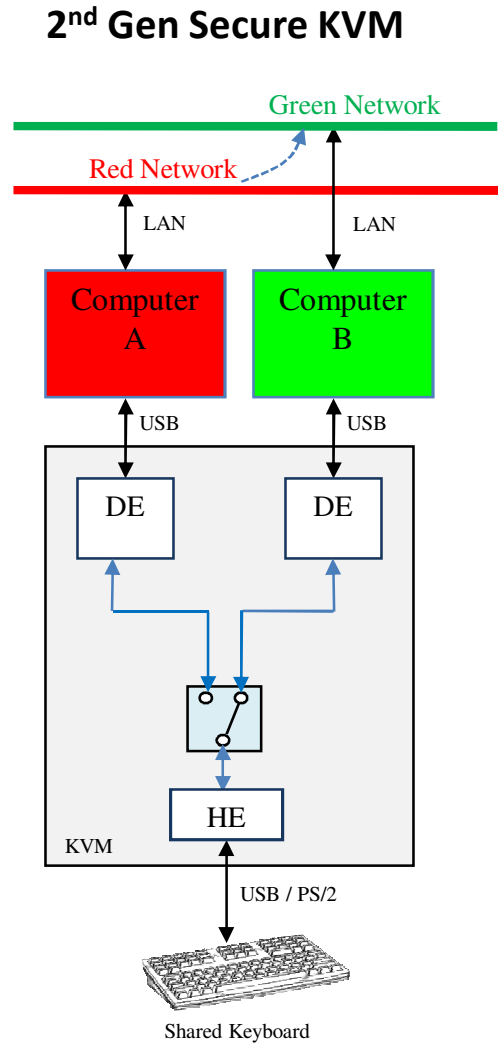
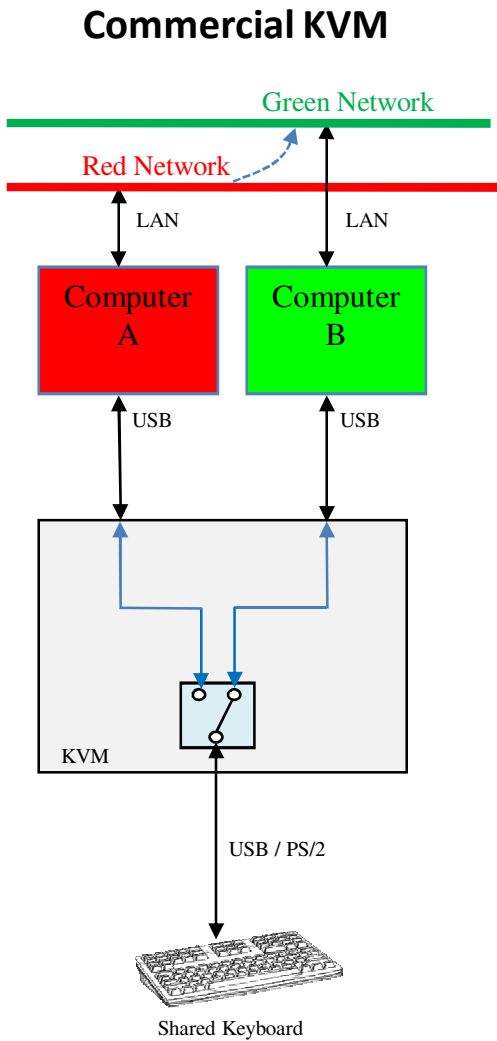
How often attacks like that happens?

No one knows...

Most of them are undetected.

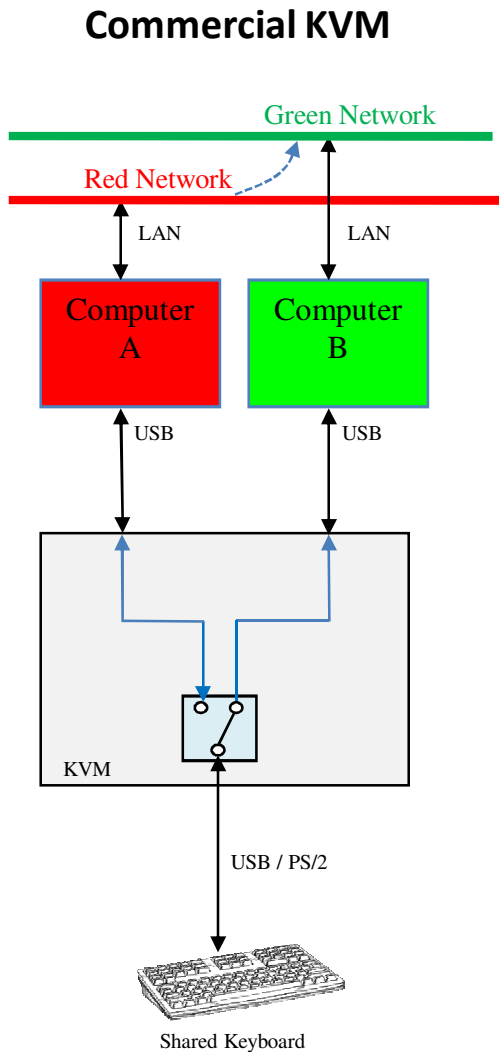
The common assumption in the Cyber defense community is: *"If the can – they will!"*

Keyboard-KVM Vulnerabilities



HE = Host Emulator
DE = Device Emulator

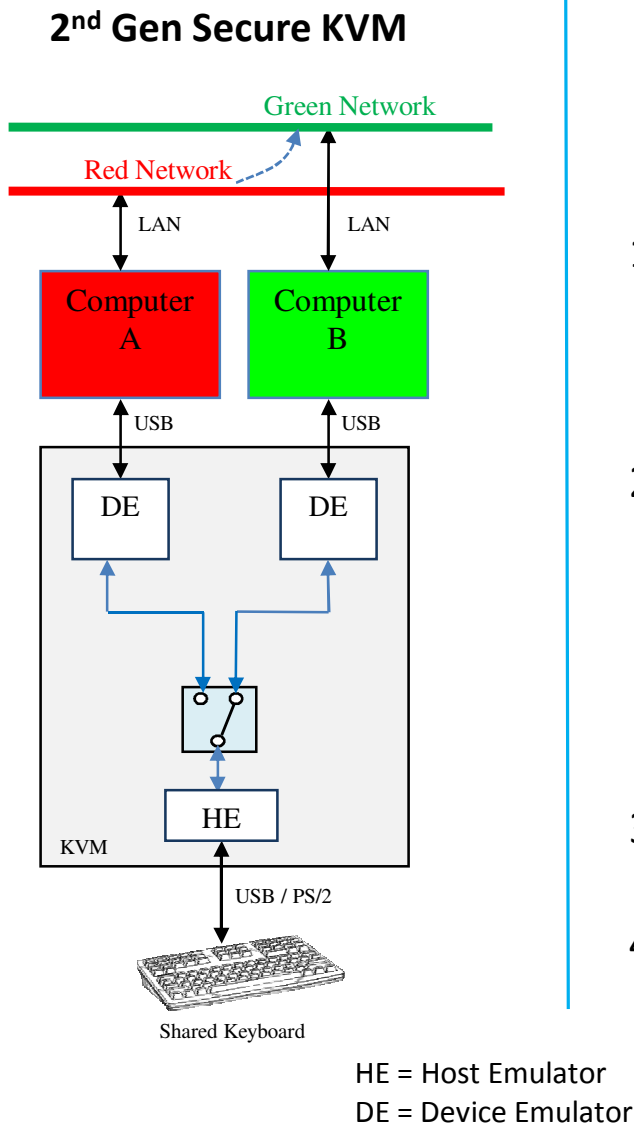
Keyboard-KVM Vulnerabilities – Commercial KVM



Attacker uses certain residual memory effects in shared keyboard to leave messages (signal) from computer A to computer B:

1. Hostile code injected into computer A is toggling Num Lock LED on to deliver '1' and off to signal '0'
2. At each KVM switching the malicious code running in PC B analyzes these Num Lock states to receive '0' or '1'.
3. At the end of the day there are several bytes that passed. Things can go much faster at night as the KVM switching may be controlled by PC A and this will allow several Megabytes of data to leak.
4. Computer B sends the received information to the attacker using various deception techniques to prevent detection. Usually using mailboxes.

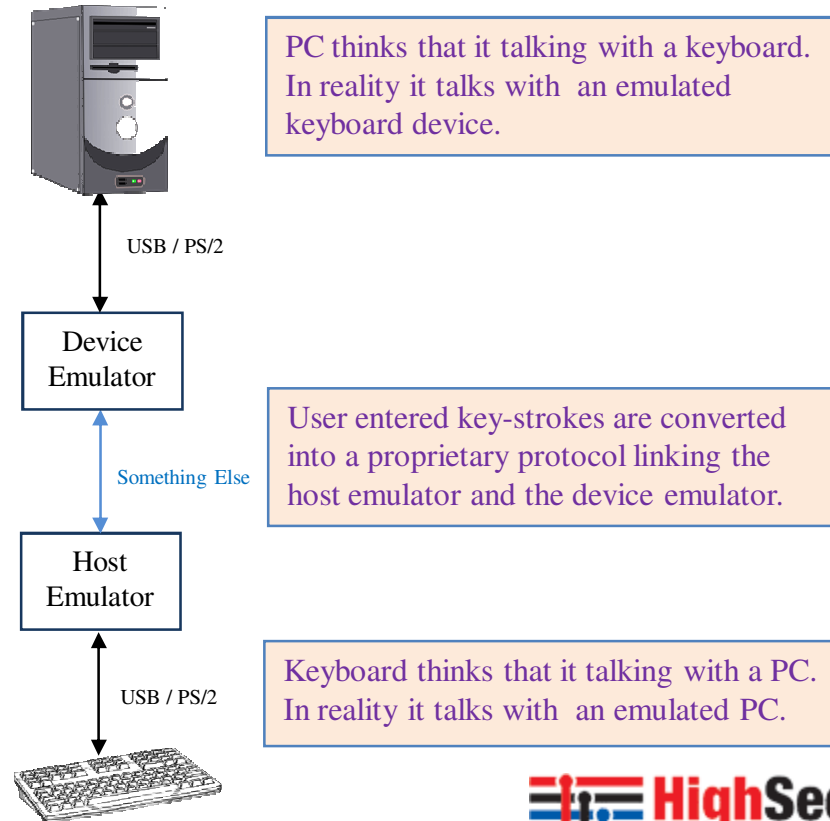
Keyboard-KVM Vulnerabilities – 2nd Gen SKVM



- Similar to the previous scenario but there the attacker is limited to a smaller number of potential signals as emulators are blocking many options. Still there are several good options.
1. Hostile code injected into computer A is toggling is saturating the keyboard controller with certain commands the to deliver '1' and not saturating to signal '0'.
 2. This saturation has a lasting effect on the keyboard that is long enough to be detected by computer B immediately after switching. At each KVM switching the malicious code running in PC B analyzes these buffers to receive '0' or '1'. Bits are collected to create bytes.
 3. At the end of the day there are several bytes that passed.
 4. Computer B sends the received information to the attacker using various deception techniques to prevent detection. Usually using mailboxes.

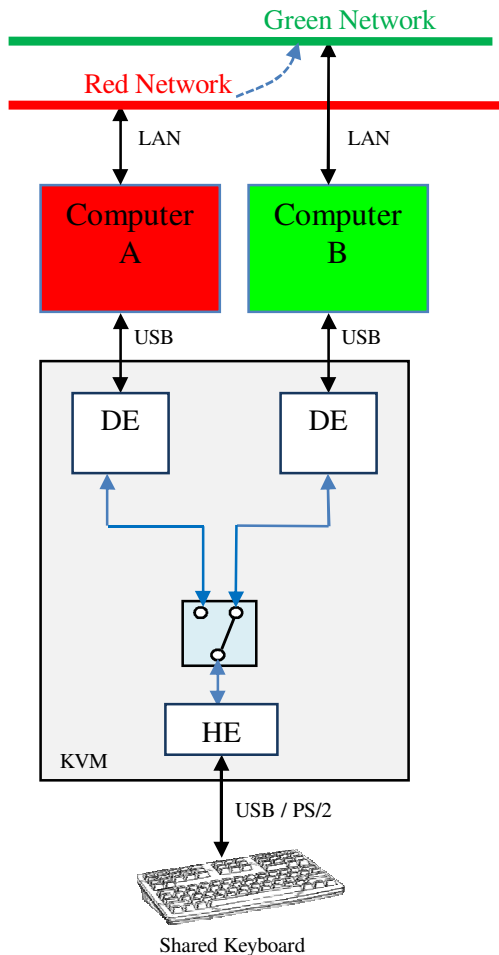
Keyboard-KVM Vulnerabilities – Security Functions

Keyboard is emulated – the connected PCs are not really communicating with a real shared keyboard. Each PC talks with its own dedicated keyboard device emulator – a microcontroller that was programmed to emulate the behavior of a real keyboard. Instructions (key-strokes) for these emulators is coming from the real keyboard through a host emulator.



Keyboard-KVM Vulnerabilities – 2nd Gen SKVM

2nd Gen Secure KVM



HE = Host Emulator
DE = Device Emulator

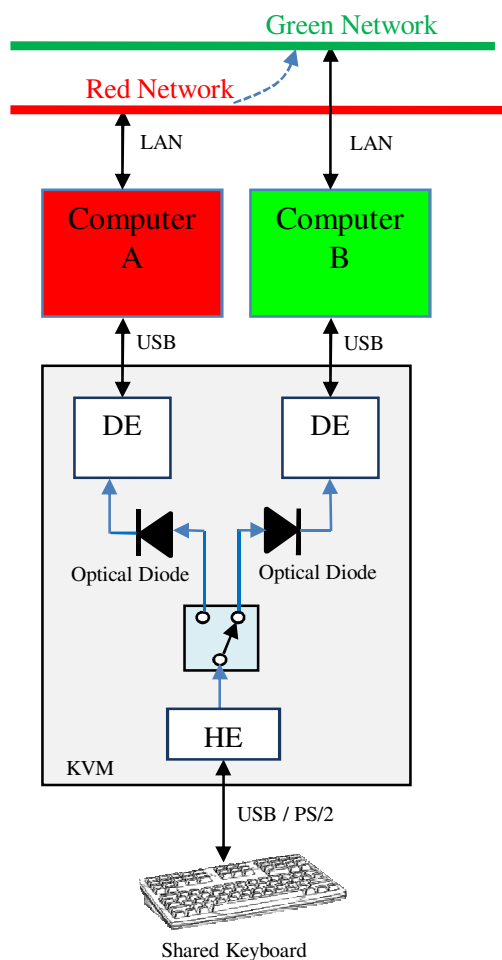
Attacker may also attempt to attack the KVM itself targeting the Device emulators. Although the device emulators are typically using Read Only Memory – there are certain methods that can be used to modify their code in real time.

Once the device emulator code was changed, the host emulator can be attacked in a similar way and the whole KVM is now a remote controlled device.

The bad thing about these attacks is that the attacker don't have to test it on your equipment. Attackers can easily buy similar equipment on-line, reverse engineer the firmware and test everything at their own lab well ahead of the real attack. The target environment is easy to build in the lab at very low effort.

Keyboard-KVM Vulnerabilities – 3rd Gen SKVM

3rd Gen Secure KVM



There is no known attack method for this KVM.

- Even if the attacker successfully reprogrammed the RED Device Emulator, he / she will never get into the Host Emulator as there is a unidirectional diode downstream.
- Same for the green Device Emulator.
- Even if somehow the Host Emulator code was modified (how?) then still information cannot leak from A to B or from B to A due to the two diodes.

Note that some Secure KVMs are using unidirectional lines without optical diodes. In these products it can be shown that if the Master controller is reprogrammed (the Device Emulator), the link can be reversed. Data may flow in the opposite direction. No optical diodes means that the unidirectional flow cannot be assured.

HE = Host Emulator
DE = Device Emulator

Keyboard-KVM Vulnerabilities

Feature	Commercial KVM	2 nd Gen Secure KVM	3 rd Gen Secure KVM
Keyboard is emulated	No	Yes	Yes
Keyboard traffic is unidirectional	No	No	Yes
Keyboard traffic is passed through optical data diodes	No	No	Yes
Keyboard emulators independently powered and isolated	No	No	Yes
Products	All non Common Criteria products,	Avocent: SC420, SC440, SC540, SC4 PDV, SC4 UAD Belkin: F1DN102U Adder: SW200xA-USB-EAL, AVSD100x Argon: Ruggedized KVM (SC440 inside)	Belkin: New Advanced DVI-I Secure KVMs

Notes:

1. In Avocent models with CAC keyboard traffic is mixed with CAC traffic!
2. Some new Adder products have unidirectional flow but no optical data diodes.

Simple KVM Tests

Most KVM vendors would not tell you what protection methods they are using. Here are some simple tests that will allow you to find out by yourself:

Does my KVM use emulators?

Connect the KVM to a computer running windows without connecting a keyboard to the KVM. Select the channel connected to the computer in the KVM. Connect keyboard and mouse directly to the PC and then look at the PC to find out what USB devices are found. If you see the extra keyboard and mouse in the KVM then they must be emulated.

Does my KVM uses isolated power to the emulators?

Repeat the above test but this time disconnect the power to the KVM. If KVM USB devices are still shown then power is isolated.

Does my KVM unidirectional?

Connect to the KVM a computer and a keyboard and check the operation of Caps Lock – if it is working then it is bi-directional.

Does my KVM have optical data diodes?

You don't have a way to test it. Just check if it is Belkin. Only Belkin uses optical data diodes.

Keyboard / Mouse KVM Vulnerabilities

Attack method	Commercial KVM	2 nd Gen Secure KVM	3 rd Gen Secure KVM
Keyboard / mouse code change attack	High Risk	Medium Risk	Low Risk
Signaling through power modulation	High Risk	High Risk	Low Risk
Mailbox attack through keyboard residual memory function, composite device, hub	High Risk	Low Risk	Low Risk
Mailbox attack through controller vulnerabilities	High Risk	Medium Risk	Low Risk
Timing signaling attack	High Risk	Medium Risk	Low Risk
Buffer overflow attack	High Risk	Low Risk	Low Risk
Social injection attack through hidden storage devices	High Risk	Low Risk	Low Risk
Enumeration mailbox attack	High Risk	Medium Risk	Low Risk
Mouse overshoot attack	High Risk	High Risk	Low Risk
KVM microcontrollers code attack	High Risk	High Risk	Low Risk

Legend:

High Risk

Medium Risk

Low Risk

Attack Methods – Keyboard / mouse code modification

The strategy

Modify peripheral device functionality so it will leak data internally (commercial KVMs) or will assist in Keyboard or Mouse mailbox leakages (in Secure KVMs).

Implementation

Attacker is aware of the keyboard / mouse microcontroller type and tested many types of attacks on same KVM in his / her lab. Attacker will run necessary code on Green computer to program required code change.

Results

Attacker controls the shared peripheral device. Signaling can be launched at night or even leakages through the KVM controller is now possible.

Prevention

Emulators + optical data diodes will block this attack.

Attack Methods – Signaling through power modulation

The strategy

Use PC A capability to switch on or off it's USB or PS/2 port power to signal '0' and '1' to computer B. In many computers power to the USB ports can be easily controlled by software.

Implementation

Attacker is using a KVM vulnerability that caused by reliance on one power source. For example: some commercial KVMs are using one USB port to power the KVM. If attacker would shutdown the other port power (computer B) then he / she can use the power control of computer A to cause the KVM to appear or disappear as a USB device in computer B. This can be easily detected by a code that the attacker runs on computer B. There are other power related methods that relies on power transient and reset timing that are too complex to be covered here.

Results

Attacker uses power effect to signal data across the KVM.

Prevention

Proper KVM power isolation design will block this attack.

Attack Methods – Mailbox attack through keyboard memory

The strategy

Use various settings that remains on the keyboard after switching to the other channels. Signal '0' and '1' by setting things on and off.

Implementation

Attacker is using keyboard designed or residual memory effect to write '0' or '1'. Every time that the KVM switches to channel B the malicious code running in A detect it and send the next bit. The use of more complex keyboard controllers with internal USB hubs or with additional multimedia keys adding more opportunities for residual memory. Wireless keyboards and mice are the worst in this aspect. There are many potential settings that may be abused including settings that would remain after device reset.

Results

Attacker uses attached keyboard or mouse to signal every time that the KVM switches over. It is not the KVM vulnerability that is being abused here – it is the keyboard and mouse vulnerability when they are being shared.

Prevention

Emulators will block this attack.

Attack Methods – Mailbox attack through controller

The strategy

Attacker recognize internal design features in the peripheral device to be abused as mailboxes for signaling.

Implementation

A good example is the Apple keyboard that allows firmware updates. Attacker may modify keyboard firmware completely to store 200 Bytes each time.

As many of the keyboards are using general purpose microcontrollers, in many cases it is possible to find there some unused mechanisms that may assist in leaking. Another example is a Cypress microcontroller used in Dell keyboards that have a set of test registers that may be abused.

Results

Same as before - attacker uses attached keyboard or mouse to signal every time that the KVM switches over. It is not the KVM vulnerability that is being abused here – it is the keyboard and mouse vulnerability when they are being shared.

Prevention

Emulators + unidirectional flow diodes will block this attack.

Attack Methods – Timing Signaling attack

The strategy

Attacker uses computer A to cause an abnormal or normal event in the keyboard or mouse. This event is initiated at an accurate timing measured against another detectable reference event. The delay sends '0' or '1' or several bit of values. Sounds complex – but in reality it is getting even more complex...

Implementation

Due to security reasons it is hard to get into the details here. Still the effect may be a much faster data leak as timing delays may be translated into 5-8 bit values in each KVM switching.

BTW – a combination of this attack with audio leakage is relatively easy to implement.

Results

Same as before – and slightly faster. Data leakage between computer A and computer B.

Prevention

Emulators will block this attack.

Attack Methods – Keyboard buffer overflow attack

The strategy

Attacker uses keyboard design flaws to store some bits in controller RAM. These bits can be read by the second computer.

Implementation

This is probably the oldest method that used primarily with commercial KVMs. Attacker floods the keyboard receive channel buffer with a long list of extra commands. In some types of keyboards some of these bits will overflow and stored in RAM. After switching Computer B will read these values through test registers in the keyboard controller.

Results

Same as before. Data leakage between computer A and computer B.

Prevention

Emulators will block this attack.

Attack Methods – Social injection attack through hidden storage devices

The strategy

Attacker exploits a USB gadget such as web camera, thumb flash drive or mouse to inject malicious code into a classified network. This is a form of social attack that assumes that if you will distribute enough gadgets around the site and employees – one of them will do the job for you.

Implementation

This is a form of social attack that assumes that if you will distribute enough gadgets around the site and employees – one of them will do the job for you. There are easy instructions how to do this in the internet.

Results

Half of the attacks described before. This is a critical condition for most of these attacks.

Prevention

If Secure KVM is the only exposed form of USB port – then the KVM will block such attack.

Attack Methods – Enumeration mailbox attack

The strategy

Attacker exploit certain device parameters that can be modified (not hard coded) in the keyboard or mouse to be used as a mailbox.

Implementation

Several bits of information can be delivered when the KVM or keyboard device is being enumerated or after reset.

Results

Same as before – and slightly faster. Data leakage between computer A and computer B.

Prevention

Emulators will block this attack. Enumeration of the shared peripheral is hidden by device emulators.

Attack Methods – mouse overshoot attack

The strategy

The attacker uses modified mouse drivers at both computer A and computer B in a way that will position the cursor at different initial position after each KVM switching from A to B. The absolute cursor position X and Y values are translated into two bytes of leaked data.

Implementation

Modified mouse driver in computer A loads the data into the mouse registers. Once KVM switches to computer B it momentarily changes mode to absolute mode and “overshoots” the cursor initial position. Initial cursor position is read by malicious code in computer B to generate 2 bytes of leaked data (X and Y position).

Results

Attacker can leak significant amount of information between computers A and B. User will be usually unaware as changes in initial position looks random.

Prevention

Emulators + optical data diodes will block this attack.

Attack Methods – KVM microcontroller code attack

The strategy

Modify KVM functionality so it will leak data internally. Moving from Device emulators to Host emulator in secure KVMs. Many commercial KVMs are using controllers that are capable of being reprogrammed in the field.

Implementation

Attacker is aware of the microcontroller type in the KVM and tested many types of attacks on same KVM in his / her lab. Attacker will run necessary code on Green computer to program required code change inside attacked KVM.

Results

Attacker controls the KVM. Signaling can be launched at night or even leakages through the KVM controller is now possible.

Prevention

Emulators + optical data diodes will block this attack.

What the DOD think about this?

Official position:

“In 2005, the Defense Science Board of the Department of Defense expressed concerns about the migration of microelectronics foundries from the United States to foreign countries and its impact on the security of microchips and microelectronic components delivered to the government and military and used in critical infrastructure [3]. If an adversary is able to gain access to a microelectronic component during the design phase, then a clandestine modification will corrupt every unit manufactured and the confidentiality, integrity or availability of any system using such a component can be compromised. Moreover, given the complexity of modern systems, such a modification can be deeply embedded into a system and difficult to detect and attribute.”

For full document see: http://www.cra.org/govaffairs/images/2005-02-HPMS_Report_Final.pdf

The message here is clear – if you can't trust your peripherals – at least connect a secure KVM that will protect you from them. Secure KVMs are made in the US and you can trust them (at least some of them).

Additional information

Much of the information discussed here was collected by agencies and it is still classified. Still additional information can be found on the internet:

Social Attacks

- <http://pentest.snsoft.com/2011/06/24/netragards-hacker-interface-device-hid/>
- <http://www.instructables.com/id/USB-Mouse-Flash-Drive-Hack/>

Keyboard attacks

- <http://www.sectechno.com/2009/08/03/vulnerability-in-apple-mac-keyboards/>
- <http://www.blackhat.com/presentations/bh-usa-09/CHEN/BHUSA09-Chen-RevAppleFirm-PAPER.pdf>
- [http://www.boomclips.com/videos.aspx/video~keyboard_attack/Keyboard Attack/Stupid videos/](http://www.boomclips.com/videos.aspx/video~keyboard_attack/Keyboard_Attack/Stupid_videos/)
- Fake chips <http://arstechnica.com/tech-policy/news/2011/06/spies-military-looking-for-hacker--backdoor-proof-circuits.ars>