Hello everyone. I hope you are having a pleasant day so far.

My name is Ryan Nolette and I am an Incident Response Consultant and Threat Researcher for Bit9.

Today I would like to show you a few new ways to do malware detection, remediation, and incident response. If we have time, we will go into forensic post mortem using Bit9 as well.

Without further ado, let's kick things off.

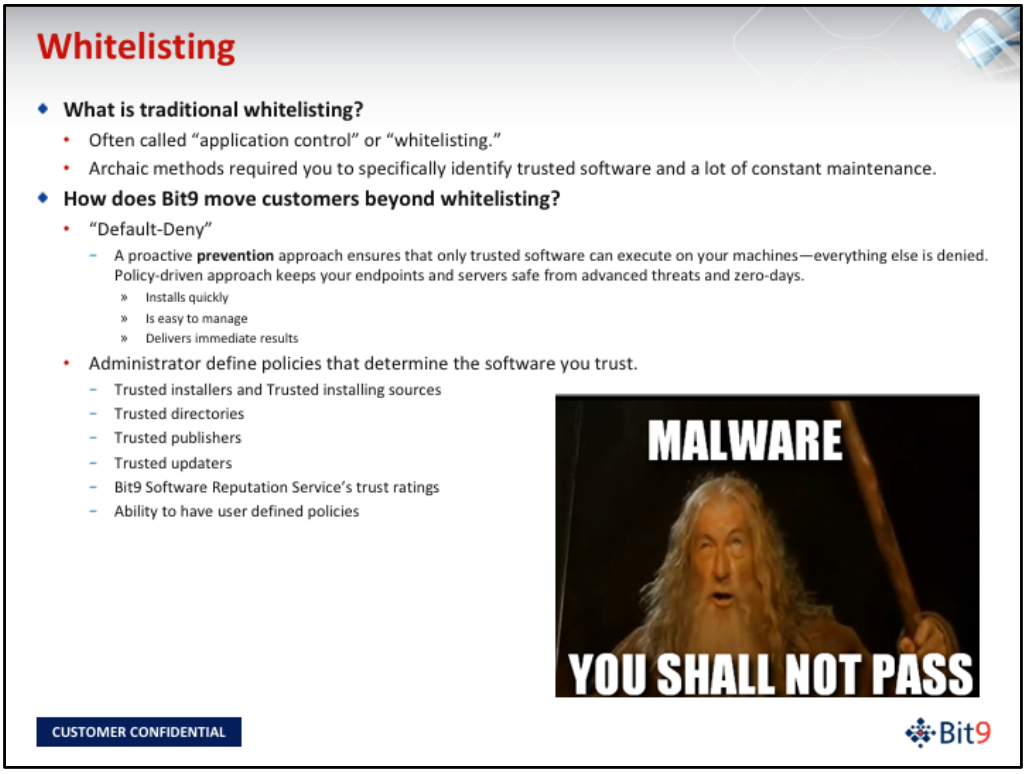
Today's Agenda

- What are a few things everyone knows we can do?
- A few things you might not know.
- Can I see an example of how this works?
  - CryptoLocker block rules leveraging Bit9
  - Zeus Incident Response leveraging Bit9
- Recap
- Questions

Bit9

Before we go into things that are new and shiny, I would like to quickly cover 3 areas everyone already knows about and that Bit9 does very well.

They are whitelisting, basline drift, and granular enforcement modes.



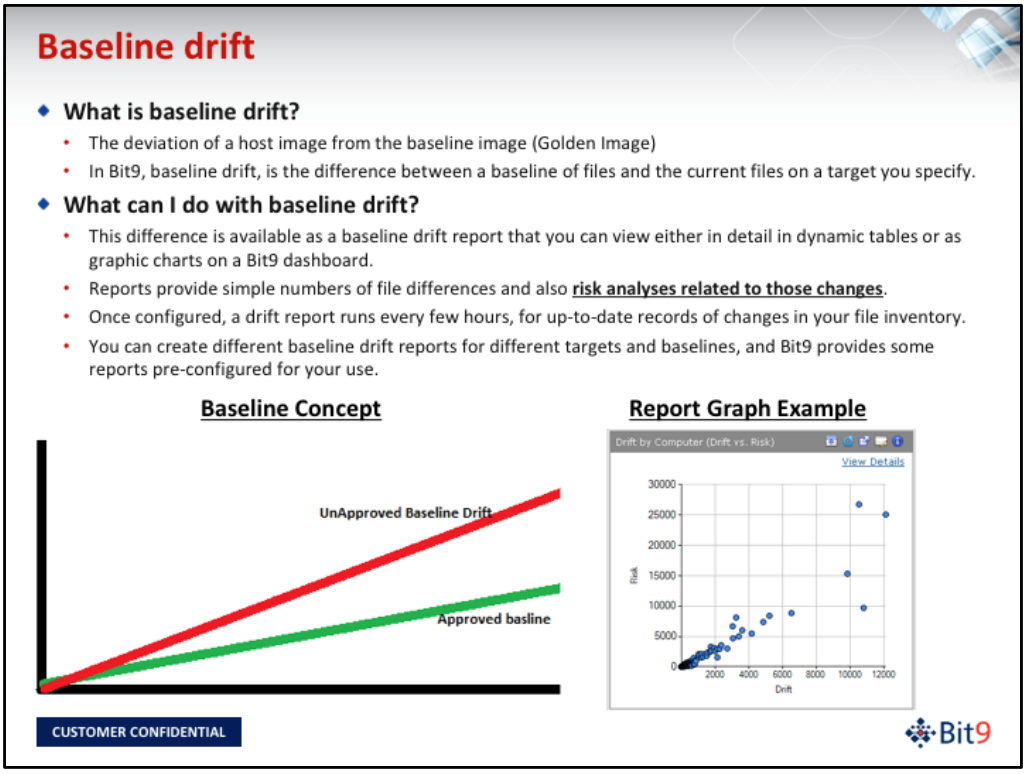
I can hear what a lot of you are thinking, "why is whitelisting a separate topic? Isn't everything that Bit9 does just to support their whitelisting ability?"

No. no it isn't.

White listing is just a portion of what Bit9 can do for you. Whitelisting is a dirty word. It isn't a dirty word because it doesn't work, on the contrary, it is a dirty word because it does the dirty work, and does it well.

Bit9's approach to whitelisting is policy driven, just like your enterprise domain, just like your AV solutions, just like anything else in life that is well structured and built to scale with demand.

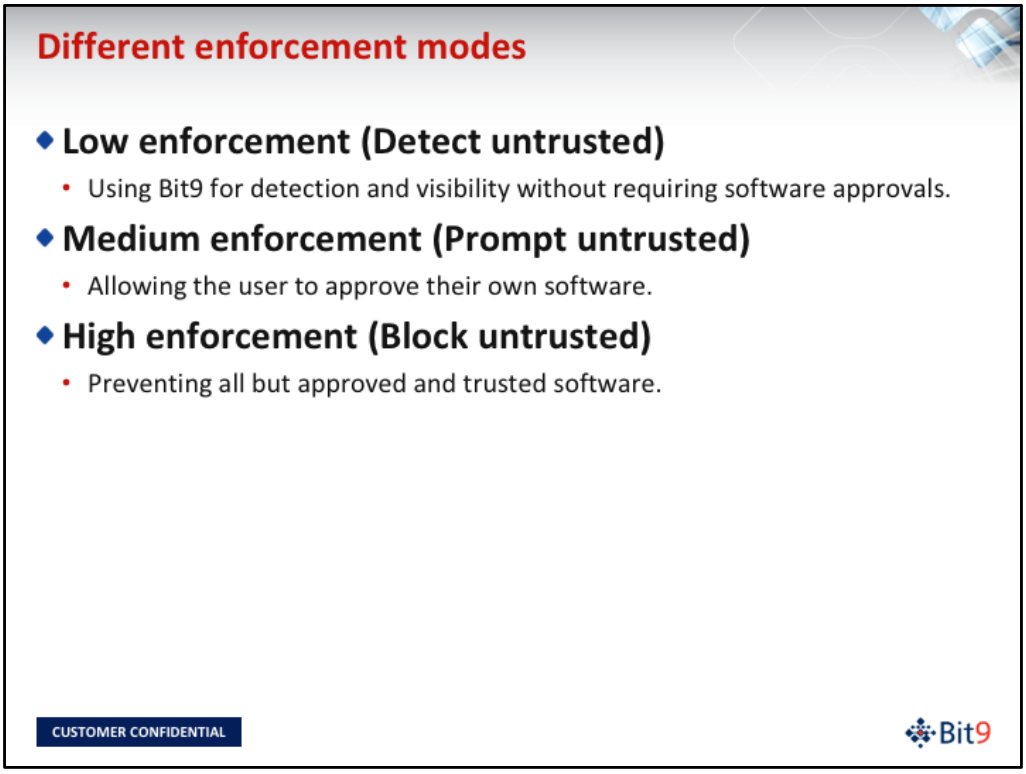
Above are a few things that separate us from traditional whitelisting the same way that fine wine is not a grape juice box.



Another feature of Bit9 that is rarely spoken about in length our the baseline drift monitoring.

What is baseline drift? Baseline drift is the deviation of a host image from the golden image as specified by you.

What is a golden image? It is the custom os installation that most enterprises build to reimage endpoints, create VDI's or VM's, or reimage Servers.

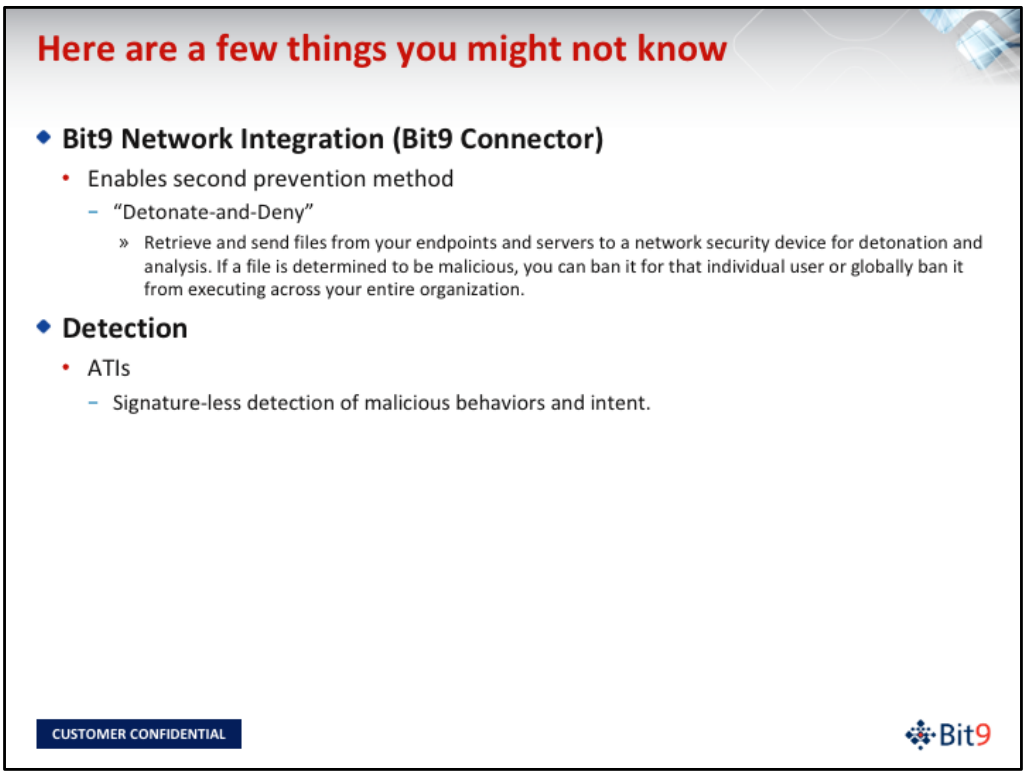


And last but not least of this category is the ability to make multiple policies for every group, business unit, or endpoint type in your environment.

The combinations are really only limited by your imagination.

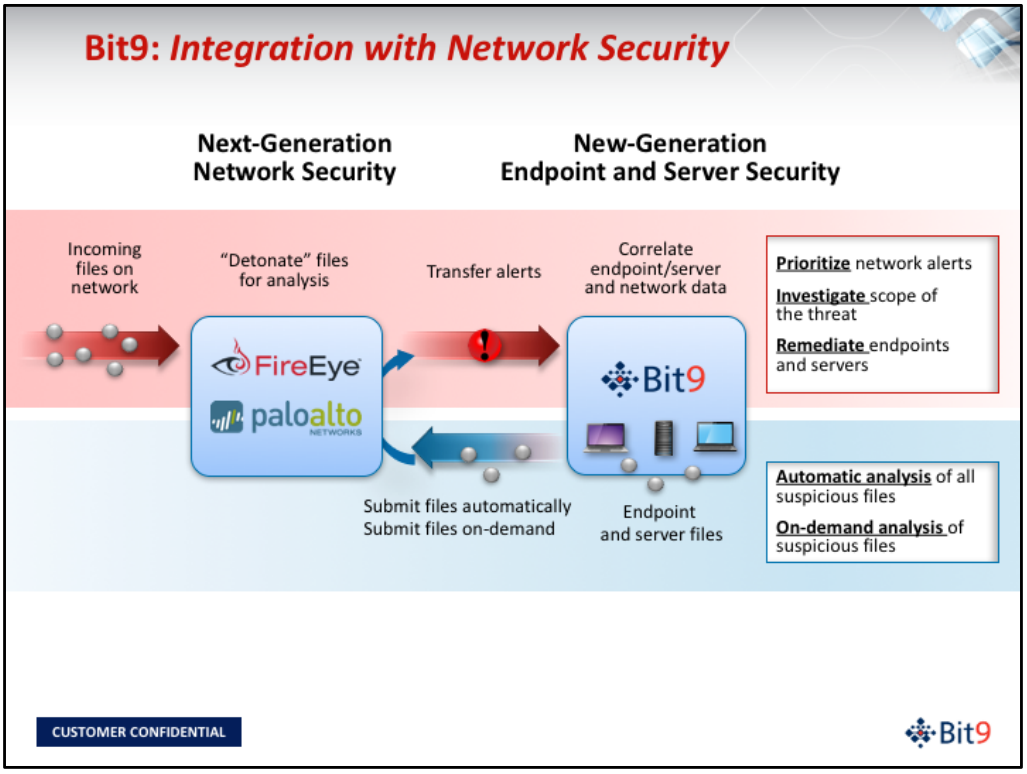
Though we at Bit9 always recommend you place your endpoints in High enforcement mode, we know that it is not always a possibility right away.

Because of these circumstances, we provide multiple enforcement levels, enforcement exception rules, and feature enhancements such as connectors and detection which we will be talking about shortly.



In the past year, Bit9 has released these 2 new enhancements which you can help leverage in your environment for either proactive blocking and protection or reactive detection, remediation, and prevention.

As I am sure you all have heard of these, I will keep this short as well and then we can dive into the meat of this presentation, how to use these features for Incident Response and malware triage.



Connectors are awesome. Let's just get that out of the way right now. They are only getting better and more feature robust with every version and it is already one of our most popular features.

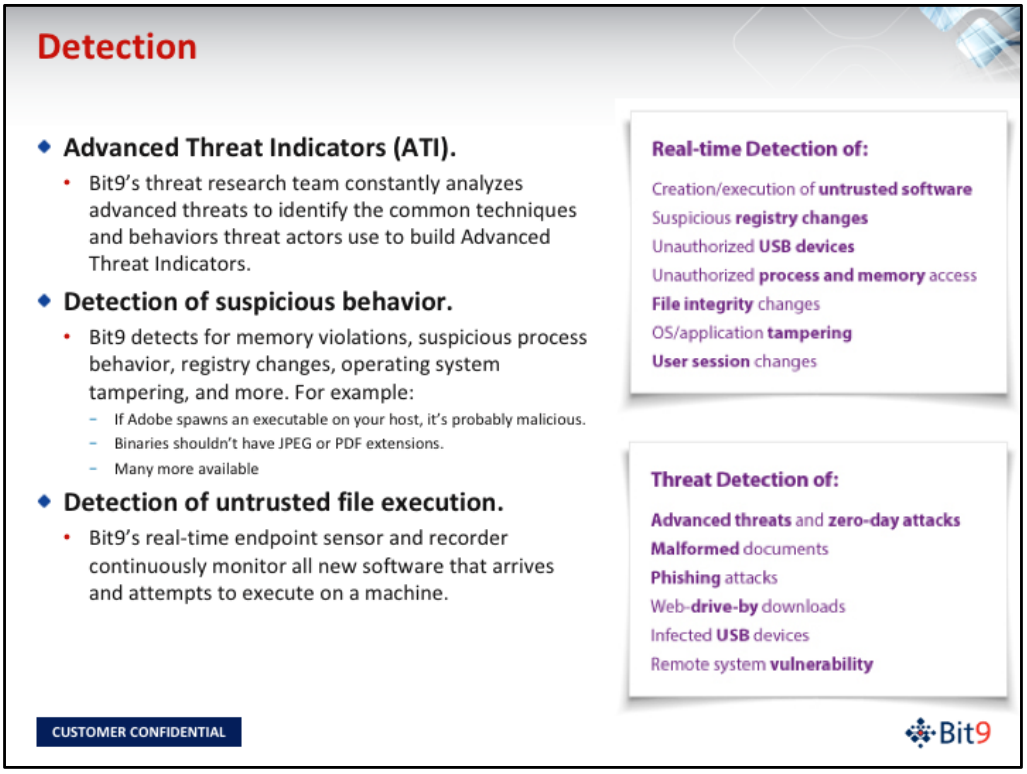
I'm a big fan of this because before I came to Bit9, I wrote something very similar to this manually for my Fireeye installation. Let's just say my coding attempts are better left uncommented to keep my peers from crying, but it worked……ish

The connector method as shown above is, as I already said, pretty awesome. It allows you to send any file you want to fireeye or palo alto for detonation, and then to receive that detonation information back in the bit9 console to either be manually interpreted or to run against automated logic to auto ban files based on your specified criteria. It's all your preference.

It also allows direct input from those network appliances to auto ban files in your environment based on rules you can setup. it's like have a whole team of soc analysts and security administrators at your finger tips.

Within seconds of a client downloading a malicious file, it can be banned globally in your environment. This is how automated prevention and protection should be done. And that isn't my opinion as a Bit9 employee, but my opinion as a 10 year veteran of Incident response, SOC building, and Security architecting.

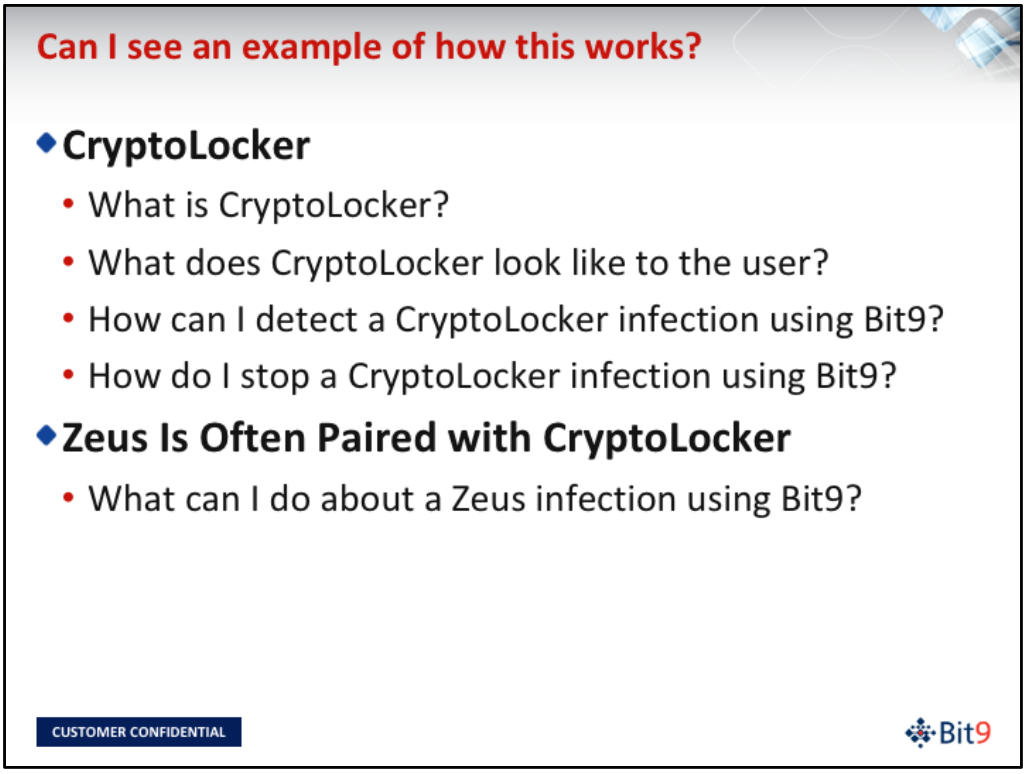
Use a setup like this, and you wont have much need to my IR presentation following.



So what do you do if you don't want to spend millions on network security appliances to pair with Bit9 in your security stack? Grab our free detection enhancement and turn it on.

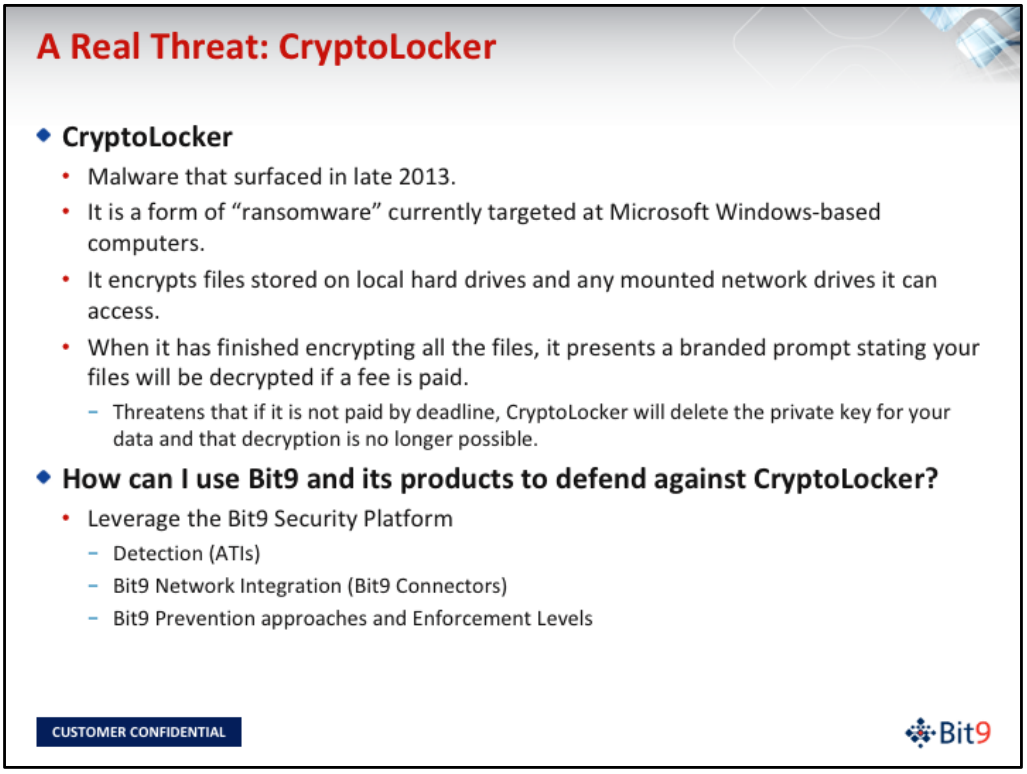
Now. no really, do it. pretty please? I mainly say that because I'm on the team that researches and develops the ATI's for the detection enhancement and I want you all to see how shiny it is.

Will the detection enhancement replace a full stack setup including connectors to network appliances? No. but it will give you actionable intelligence based on endpoint activity and help you find malicious and suspicious behavior in real time on your endpoints that AV and network appliances cannot do.



Now that we have gone over a few of the features we will be using today, let's talk about a pretty hot topic. CryptoLocker. We are also going to discuss Zeus today. Why both? Because they are being paired together as a dual threat to your enterprise.

Read above questions

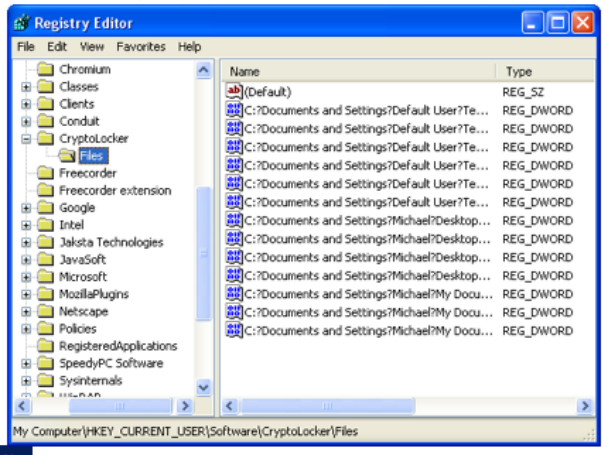


talk about cryptolocker a bit

This is all the user see's when they get infected with CryptoLocker. The problem is, by the time they have seen this it is already too late.

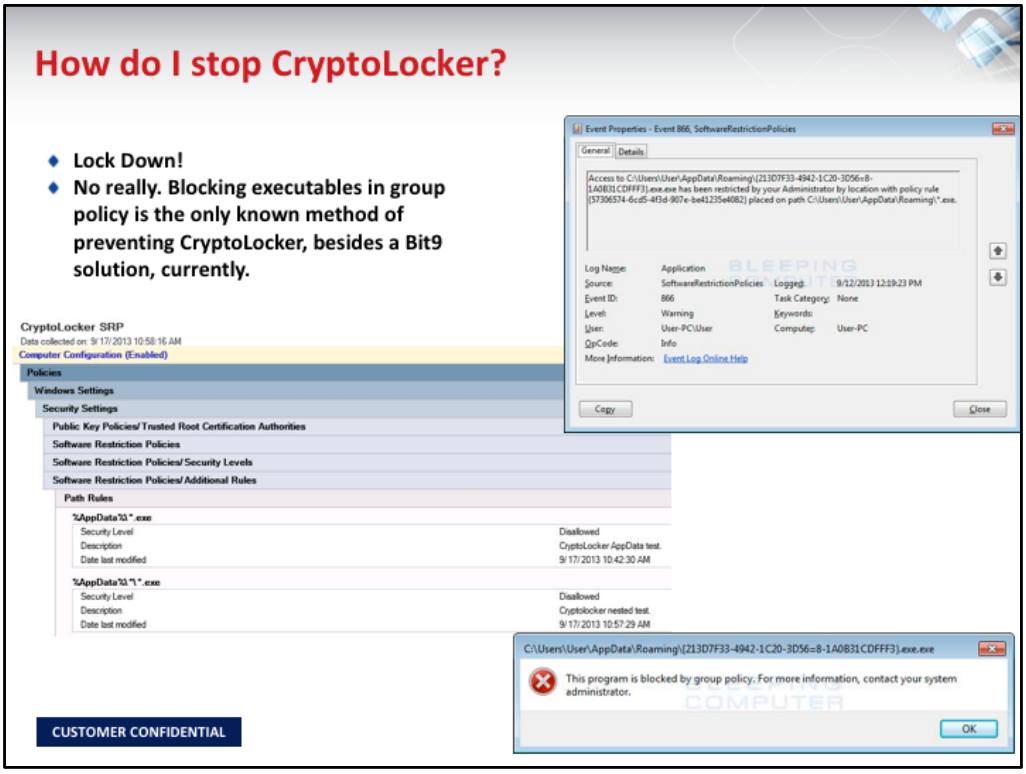
What does CryptoLocker do?

- **For each file that is encrypted, a resulting registry value will be created under this key: HKCU\Software\CryptoLocker\Files**
- **Once the infection is active on your computer it will scan your drives (local & network) and encrypt the following types of files with a mix of RSA & AES encryption:**
  - *.odt, *.ods, *.odp, *.odm, *.odc, *.odb, *.doc, *.docx, *.docm, *.wps, *.xls, *.xlsx, *.xlsm, *.xlsb, *.xlk, *.ppt, *.pptx, *.pptm, *.mdb, *.accdb, *.pst, *.dwg, *.dxf, *.dxg, *.wpd, *.rtf, *.wb2, *.mdf, *.dbf, *.psd, *.pdd, *.eps, *.ai, *.indd, *.cdr, ????????.jpg, ????????.jpe, img_*.jpg, *.dng, *.3fr, *.arw, *.srf, *.sr2, *.bay, *.crw, *.cr2, *.dcr, *.kdc, *.erf, *.mef, *.mrw, *.nef, *.nrw, *.orf, *.raf, *.raw, *.rwl, *.rw2, *.r3d, *.ptx, *.pef, *.srw, *.x3f, *.der, *.cer, *.crt, *.pem, *.pfx, *.p12, *.p7b, *.p7cThe

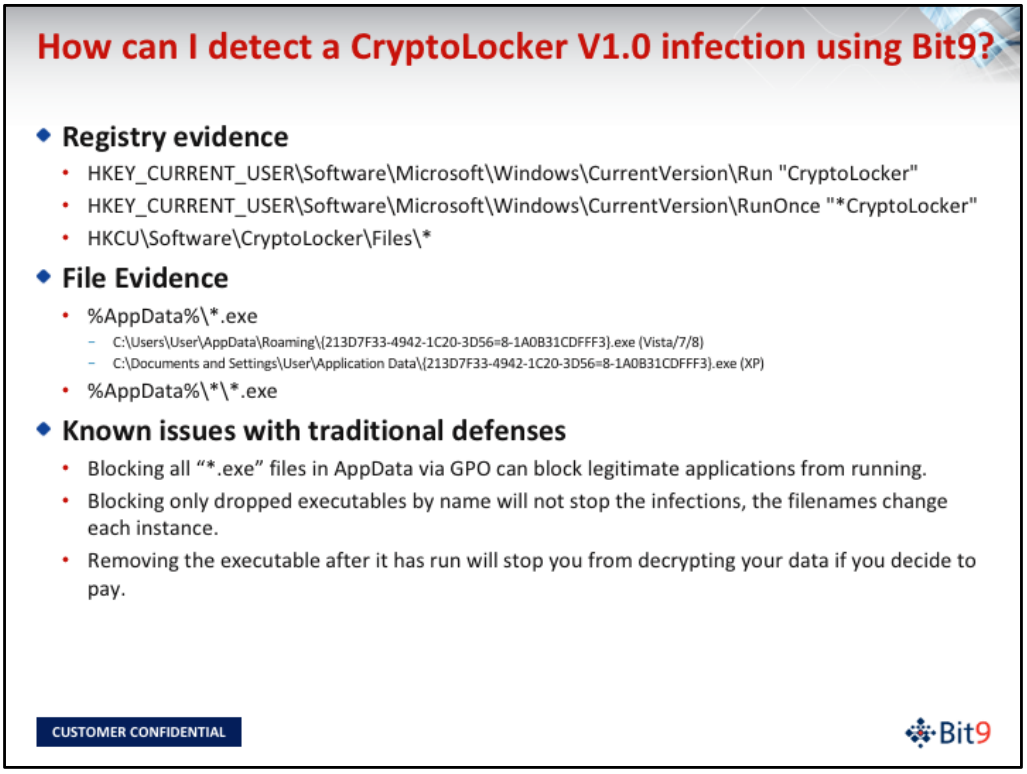


Bit9

Why is it already too late? Because once Cryptlocker starts running, it encrypts everything with the file extensions you see above, more too probably but in the samples I analyzed, there were only these extensions being encrypted.

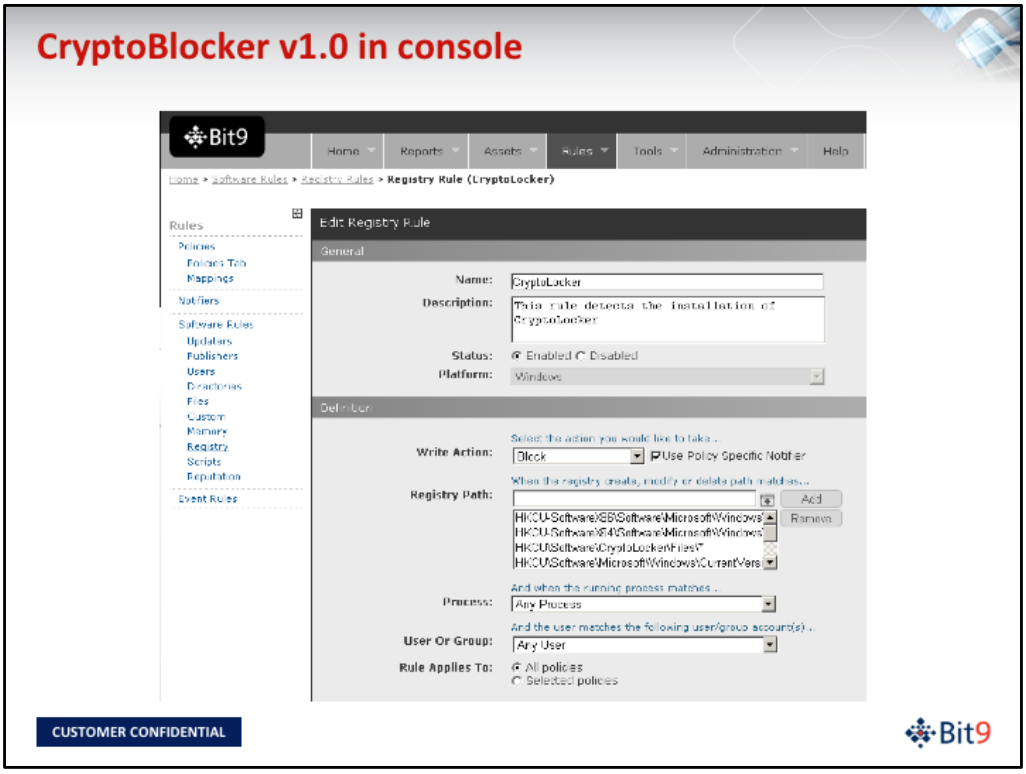


There are only 3things you can do about cryptolocker. You can have a preventive approach like Bit9 or another method block executables in specific locations (which can cause legitimate applications to stop working), restore from backups after the fact and deal with the data loss, or you can do nothing and cry.

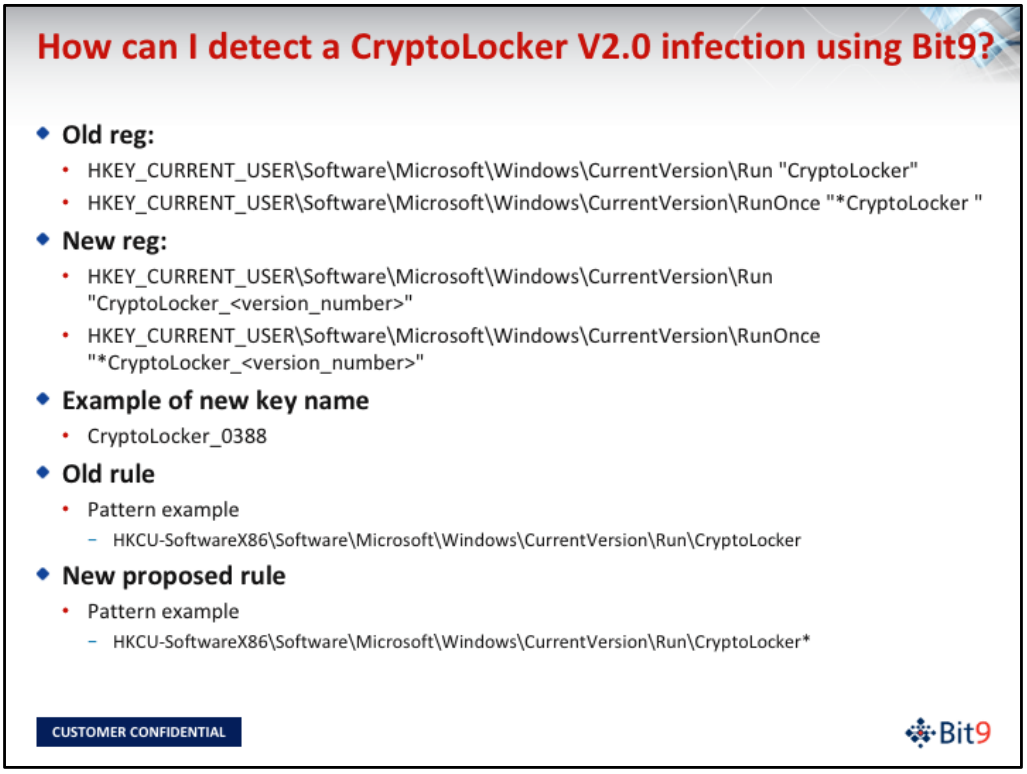


What do we know about how cryptolocker works along all installations? These persistent behaviors and endpoint artifacts.

Why does this matter? If a piece of malware is dependent on constant to succeed, we have something we can take away from it to make it fail. Similar to pulling away the lowest blocks in the jenga tower, the application will fail. This is also similar to installation an application without it's dependencies. The installation will fail and the application will not run.

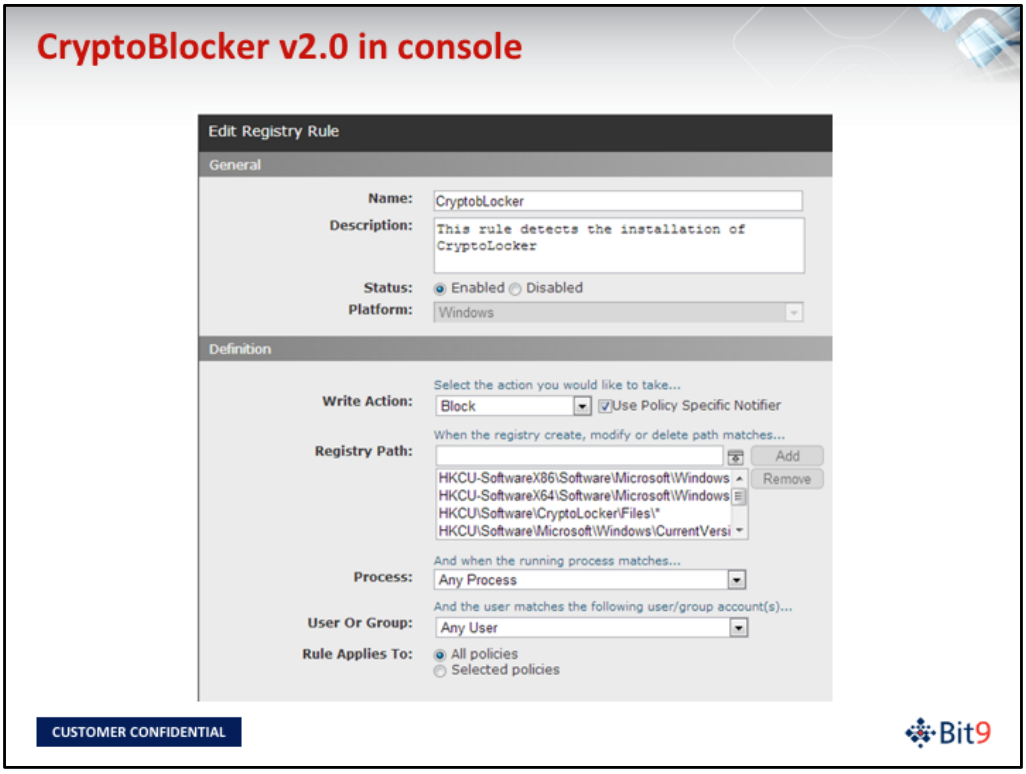


This is what our blocking rule looks like in the console.



Just adding a wildcard to the end of the registry value allows for us to block both versions of CryptoLocker from completing installation and running.

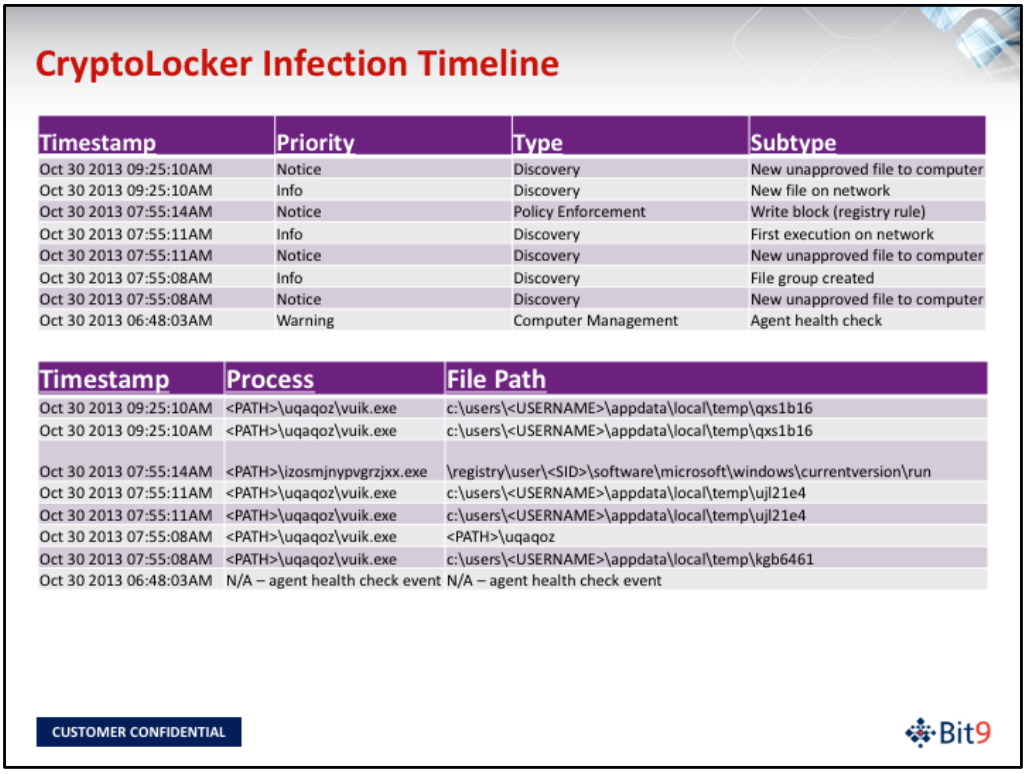
I also suggest removing all files that these rules show as they are the executables for CryptoLocker.



Version 2.0 looks an awful lot like version 1.0 doesn't it? This is because the authors did not change the malware's logic, and because of this, we can use the same technique as before to disable it.

The previous few slides detailed the method that we use to prevent CryptoLocker from running. Above is a raw XML output of the CryptoBlocker rule directly from the console.

Does it work? Yes it does. As you can see here, we are successfully blocking CryptoLocker from installing over and over again. Want to see more?



CryptoLocker Infection Timeline

Timestamp	Priority	Type	Subtype
Oct 30 2013 09:25:10AM	Notice	Discovery	New unapproved file to computer
Oct 30 2013 09:25:10AM	Info	Discovery	New file on network
Oct 30 2013 07:55:14AM	Notice	Policy Enforcement	Write block (registry rule)
Oct 30 2013 07:55:11AM	Info	Discovery	First execution on network
Oct 30 2013 07:55:11AM	Notice	Discovery	New unapproved file to computer
Oct 30 2013 07:55:08AM	Info	Discovery	File group created
Oct 30 2013 07:55:08AM	Notice	Discovery	New unapproved file to computer
Oct 30 2013 06:48:03AM	Warning	Computer Management	Agent health check

Timestamp	Process	File Path
Oct 30 2013 09:25:10AM	<PATH>\uqaqoz\vuik.exe	c:\users\<USERNAME>\appdata\local\temp\qxs1b16
Oct 30 2013 09:25:10AM	<PATH>\uqaqoz\vuik.exe	c:\users\<USERNAME>\appdata\local\temp\qxs1b16
Oct 30 2013 07:55:14AM	<PATH>\izosmjnypvgrzjxx.exe	\registry\user\<SID>\software\microsoft\windows\currentversion\run
Oct 30 2013 07:55:11AM	<PATH>\uqaqoz\vuik.exe	c:\users\<USERNAME>\appdata\local\temp\ujl21e4
Oct 30 2013 07:55:11AM	<PATH>\uqaqoz\vuik.exe	c:\users\<USERNAME>\appdata\local\temp\ujl21e4
Oct 30 2013 07:55:08AM	<PATH>\uqaqoz\vuik.exe	<PATH>\uqaqoz
Oct 30 2013 07:55:08AM	<PATH>\uqaqoz\vuik.exe	c:\users\<USERNAME>\appdata\local\temp\kgb6461
Oct 30 2013 06:48:03AM	N/A – agent health check event	N/A – agent health check event

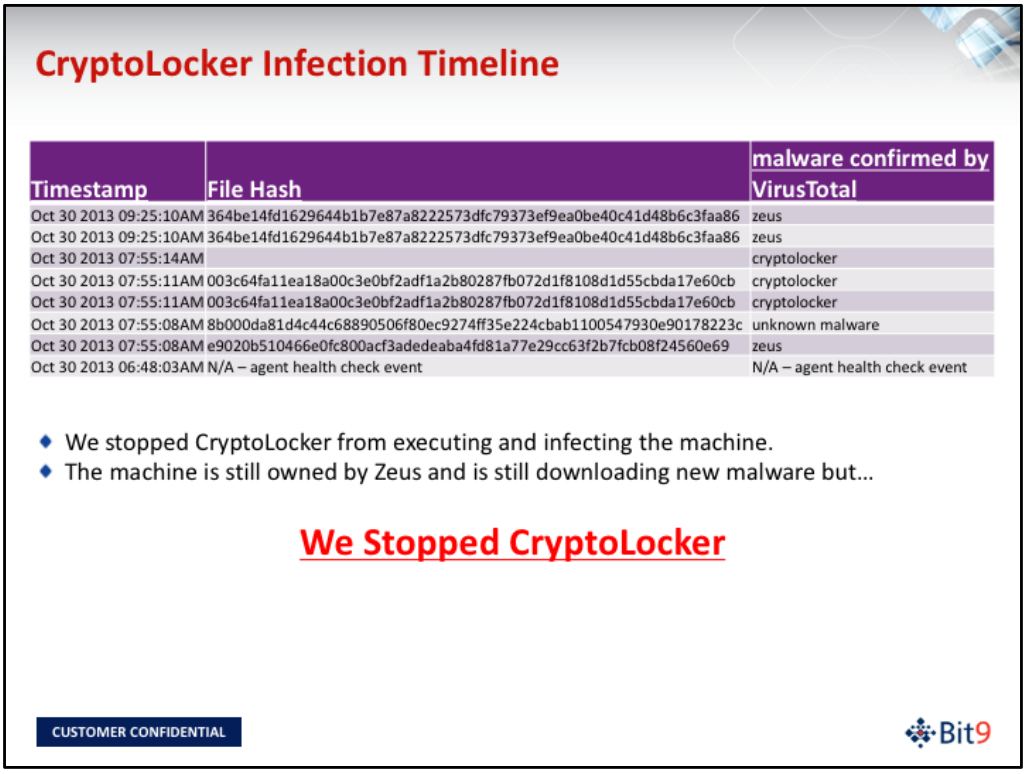
CUSTOMER CONFIDENTIAL

Bit9

The next few slides are sanitized customer data. Since the columns are so long in the console, I have taken that data and directly placed it into these tables. The data has not been changed, only sanitized.

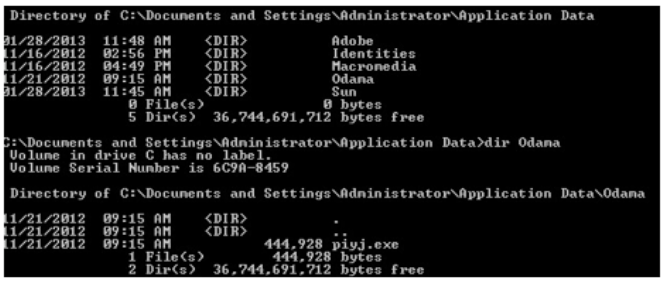
To keep a constant across all tables, I have the same timestamp column represented across each table.

What you can see from these columns is an unreleated health check on the host, then an hour later, Zeus gets dropped on the host, brings down cryptolocker and tries to run and install cryptolocker. You can see the attempt being blocked.



To clarify, we were in low enforcement mode on this host in the customer environment. Therefore they had no mechanism in place outside of their AV to protect them from Zeus, but because they applied our CryptoBlocker rule in block mode, the host was not compromised by CryptoLocker. It is however compromised by Zeus.

What can I do about a Zeus infection using Bit9?



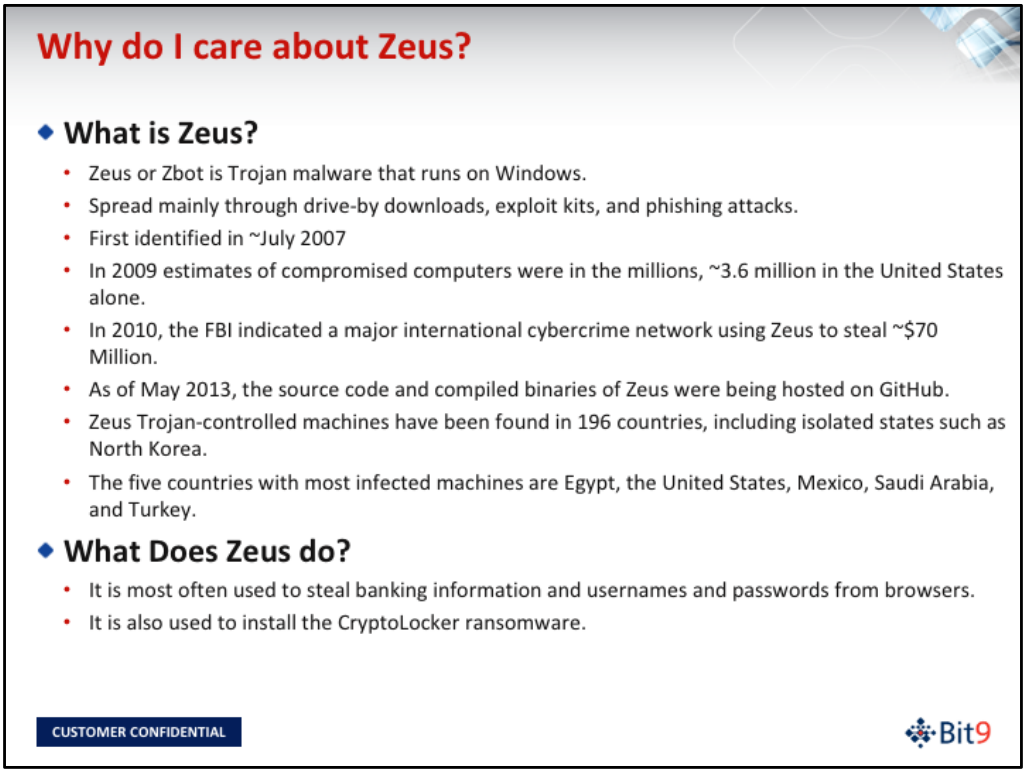
CUSTOMER CONFIDENTIAL

Bit9

Speaking of Zeus. What does a Zeus infection look like? How can we investigate a Zeus infection? How can we defend against it?

Above you see a command prompt displaying a successful Zeus infection on one of my test hosts. In the next few slides we will go in depth into the workflow of detecting and blocking a typical Zeus infection in your environment.

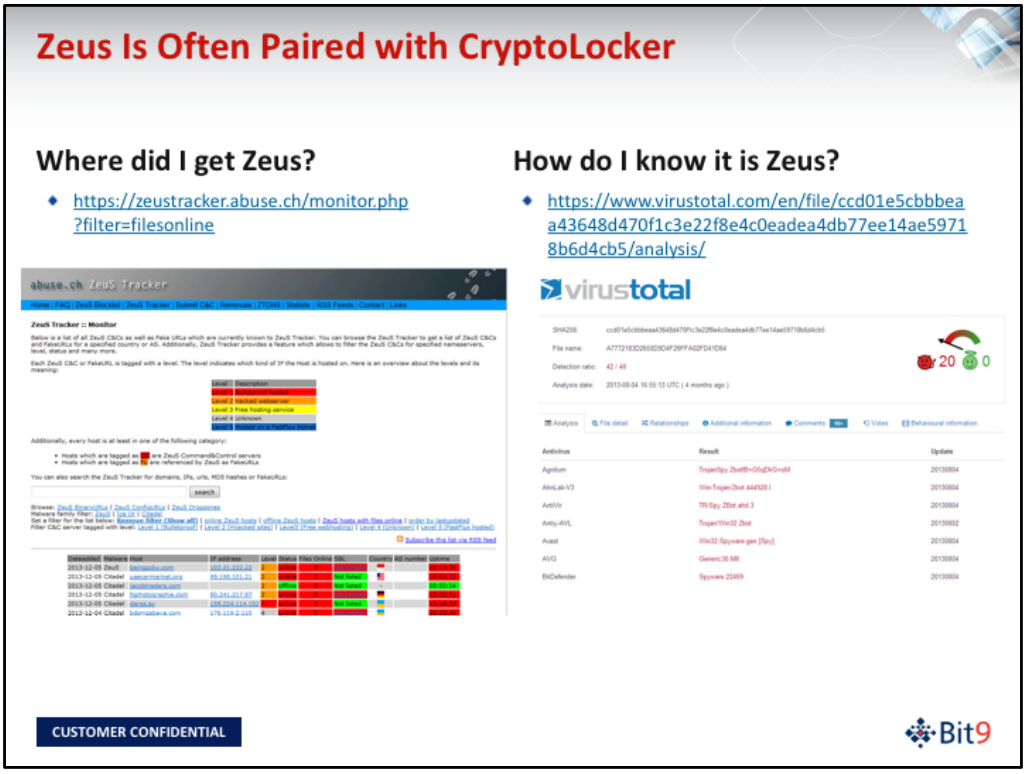
Please keep in mind. This is for Incident response and Triage only, not forensics.



Above is a few metrics about Zeus. Basically it is a piece of malicious software that gathers browser credentials and sends them to a remote server for later use.

This is the most common infection I have dealt with at financial institutions in the past 5 years.

It is now one of the most popular CryptoLocker delivery systems.



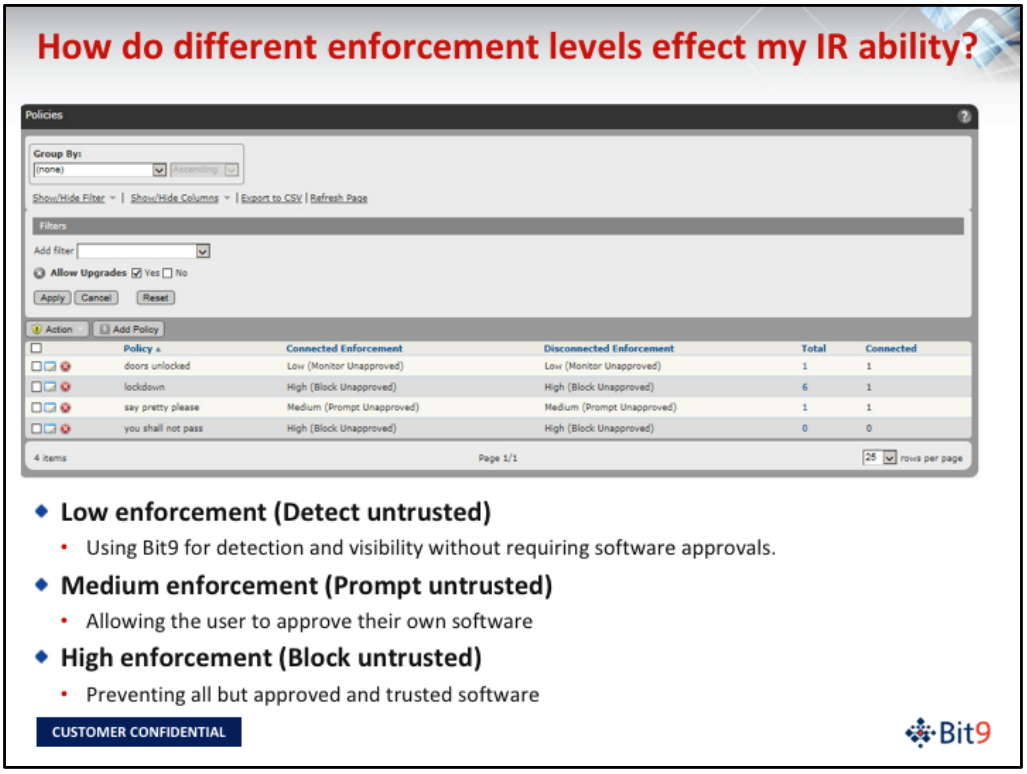
For our demo today, I am using real malware. I went to Zeus tracker, a well known source to find Zeus command servers and other Zeus related information.

I then did what most of your users do. I kept clicking on random crap until I got my host infected.

At this point I saved the executable and zipped it up for later usage. Going forward I will be depositing this zipped malware on each of my hosts in my environment and trying to execute it from there.

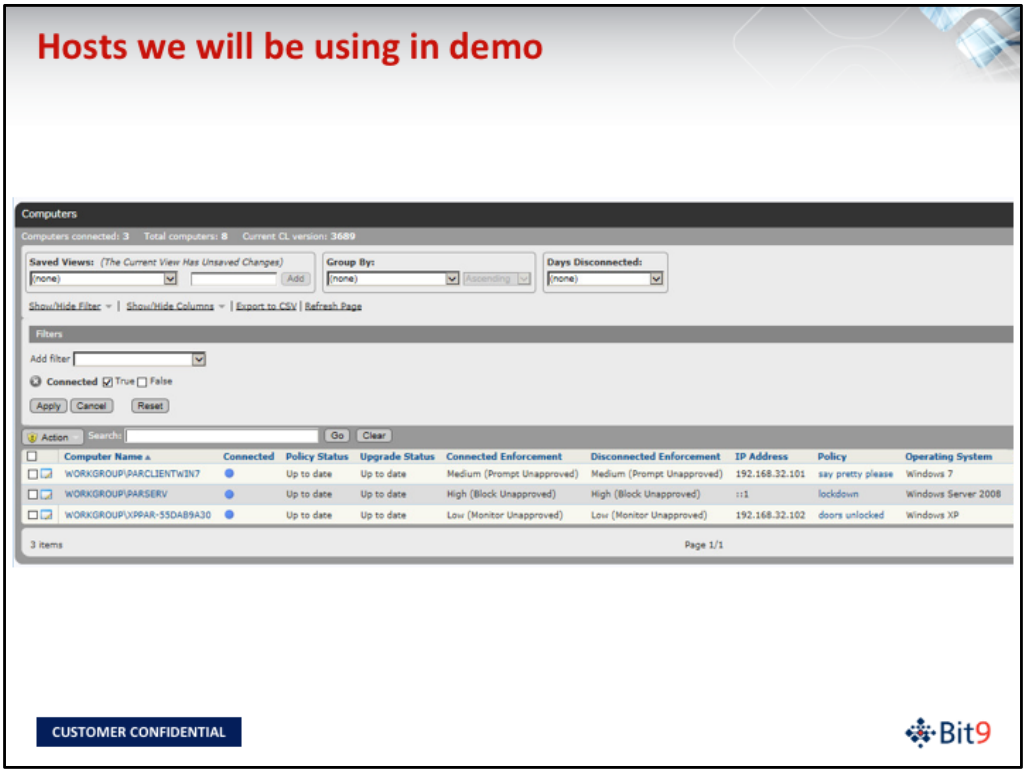
The piece of malware I was able to infect myself with on an unprotected windows XP system is to the right. Bit9 marks it with a low trust rating already but in case you wanted a third party opinion, I have uploaded it to virustotal and received a 42 of 46 rating for Zeus.

I think we can all agree, this is Zeus, this is real, and I should not click on bad links when my system is unprotected. ☺



Before we go into what Zeus looks like from the console and host, lets talk a bit about my Bit9 setup. Above are all of the policies I have in my environment. We will be using these in the demo today.

I am not using monitor mode because it is out of scope for remediation at this moment. I honestly do not use monitor mode because low enforcement gives me the same visibility as monitor mode but with the added ability of global banning.

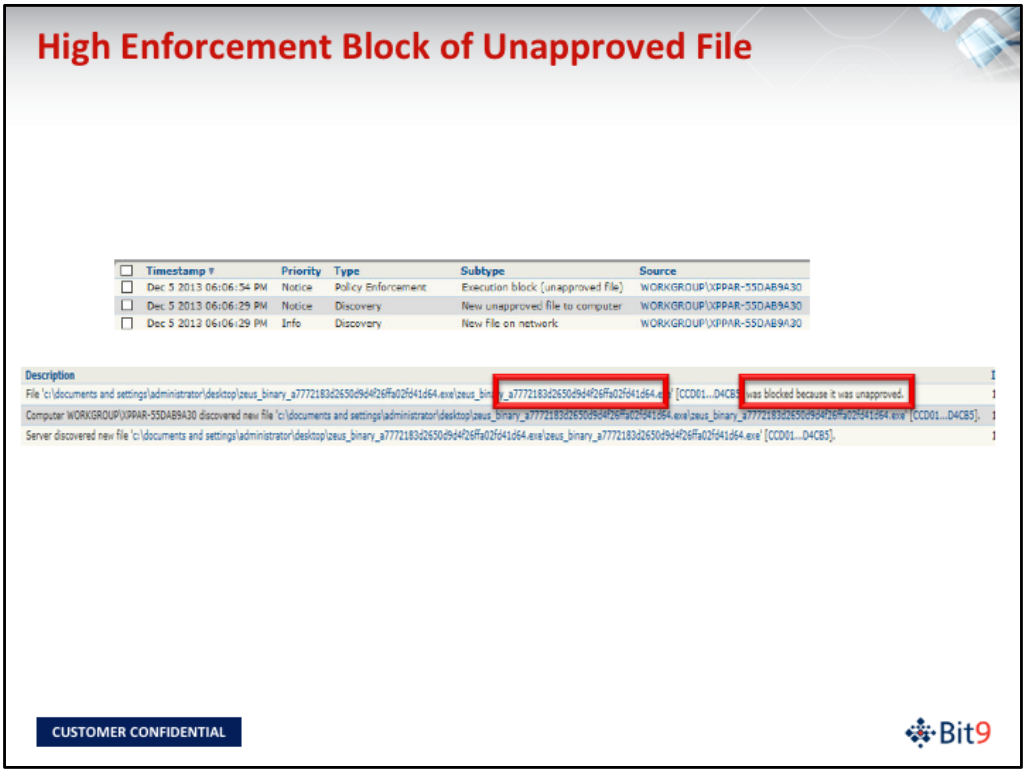


These are the hosts I am using today. I have a windows 2k8r2 server running sqlexpress and bit9. I have nothing else security wise installed. The firewall is off, and my server shares no roles on my domain besides being the bit9 server.

I have a windows7 client and a windows XP client.

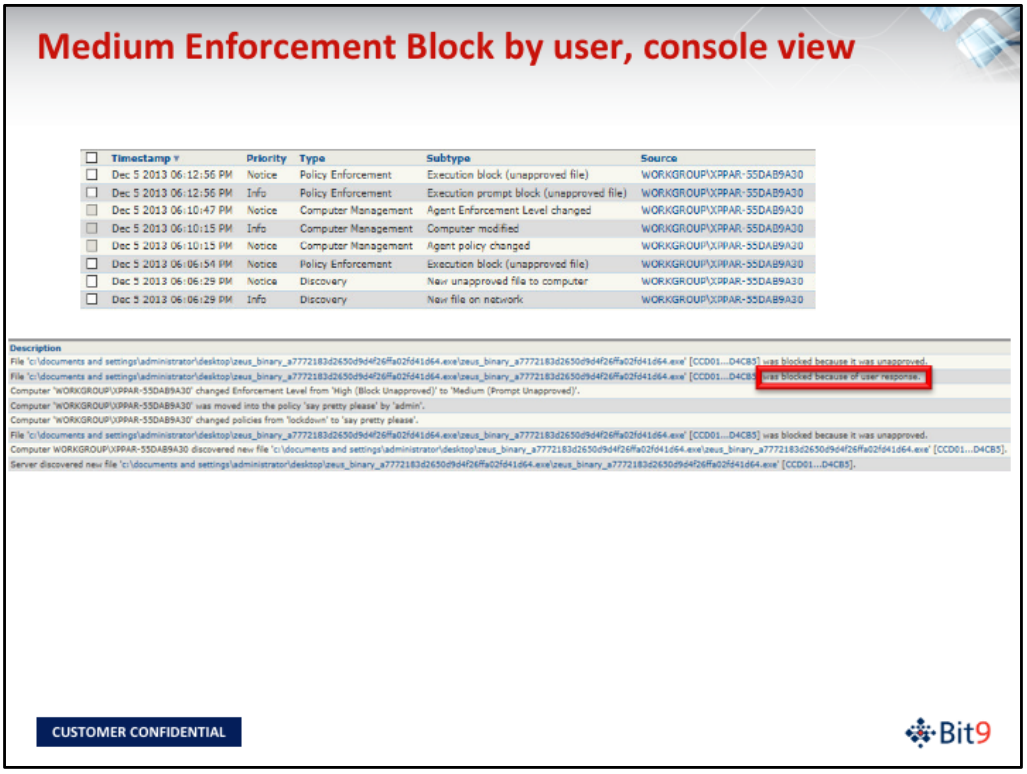
My win7 client is in medium enforcement or what I call block and ask mode. It will prompt the user for block or approve permissions for unapproved files.

My XP client is in low enforcement mode and will let anything not on my ban list install.



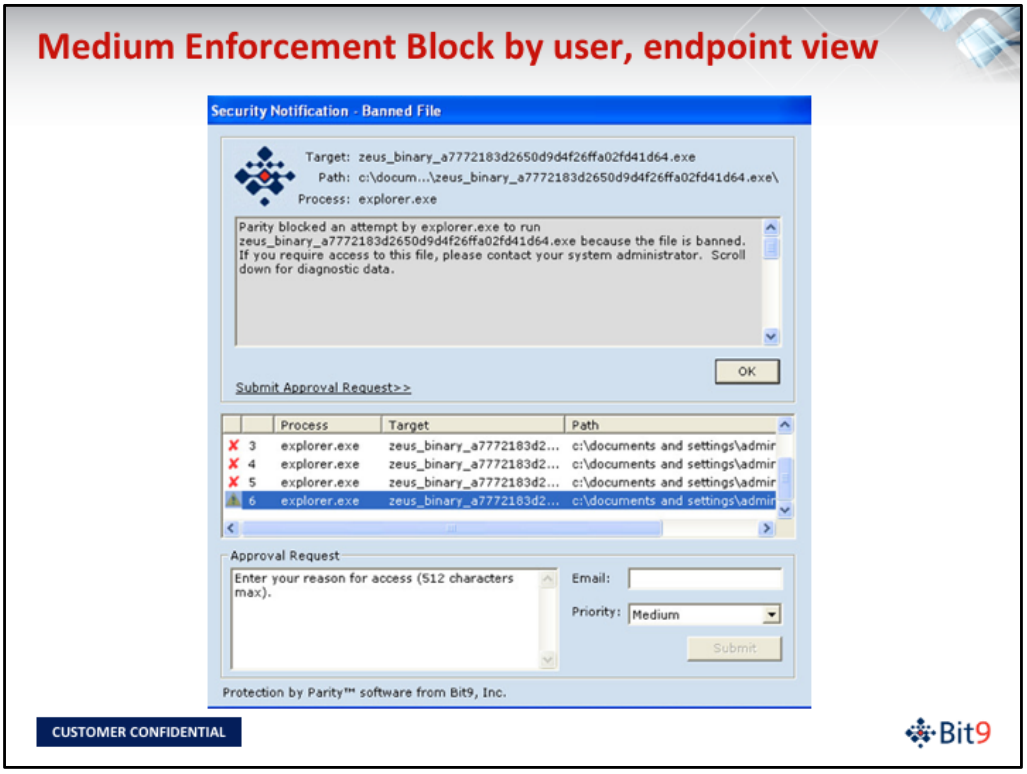
When I changed my XP machine into High enforcement mode and clicked on that bad file. As you have guessed, nothing happened. Bit9 blocked this unapproved file from executing automatically.

We can move on from here not worry about the file. Though I will say that in the next few examples we will see bad things happen because of the enforcements I will choose. I will explain everything in more detail as I go, but I wanted to point out that on this high enforcement machine, even though the file has been blocked from running, it is still on the host and should be removed. I suggest removal from each host in scope once you complete your scoping.



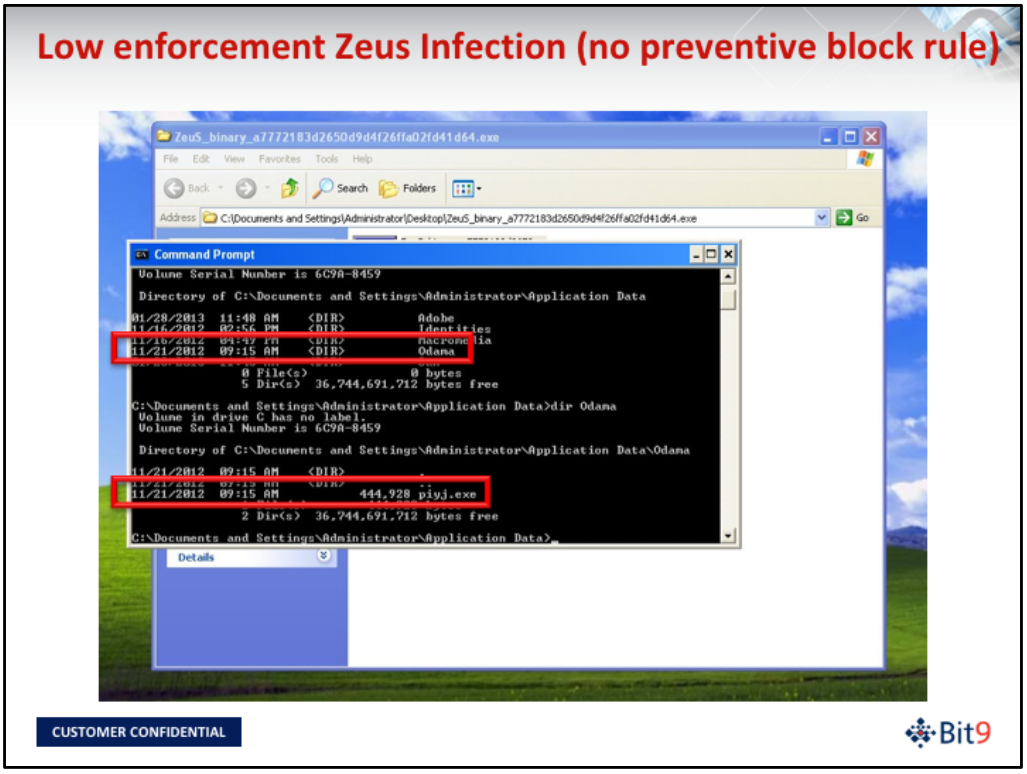
Now this is where things can get tricky. Medium enforcement mode gives the user power to block or approved unapproved files based on whatever criteria they user deems necessary. This is pretty much the same decision making that led to the file being downloaded in the first place in most cases so keep in mind that a user can still be infected in this enforcement mode if they approve everything and it has not yet been global banned.

Above you will see me click on the malware and try to execute it. I get a prompt message and I choose to block the file from running. This is what you will see in the console from that event.

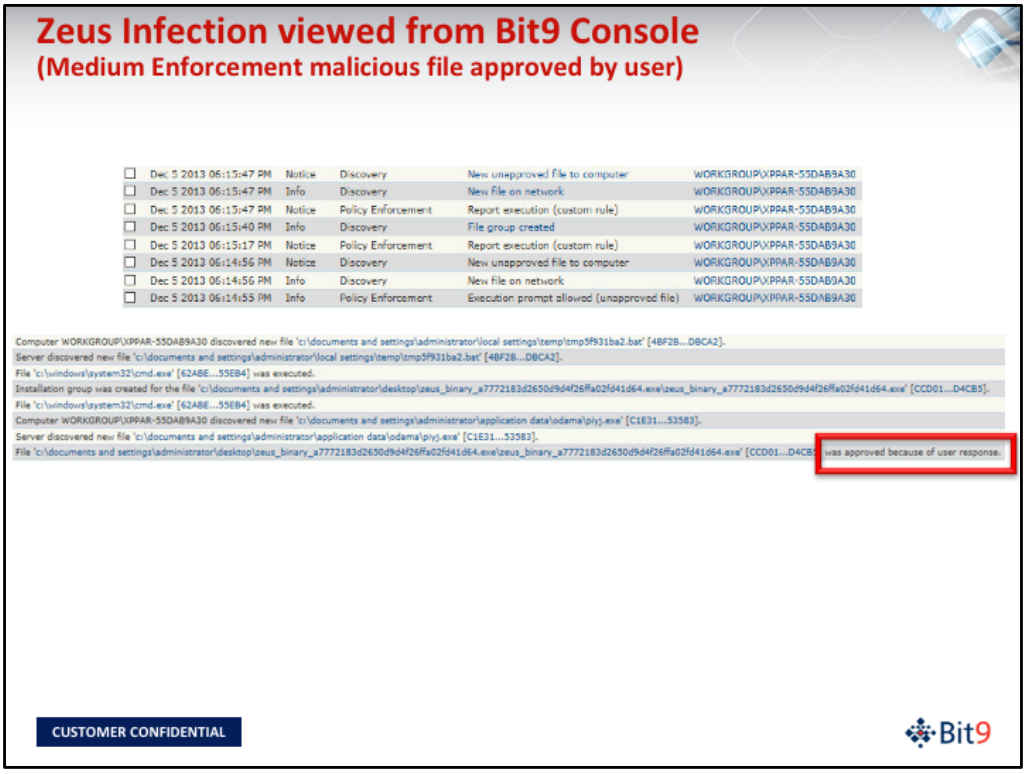


This is the flipside to the block. The user is prompted with a screen similar to the one above and can choose to block or approve the file.

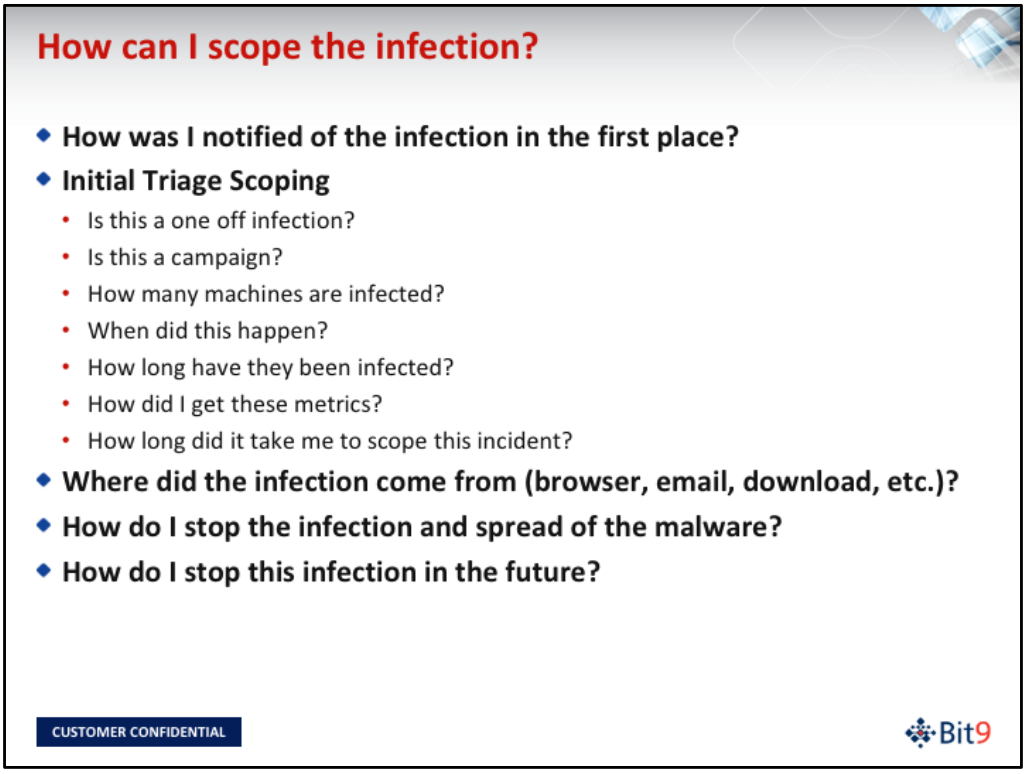
Now I have switched my Xp client into low enforcement mode with no server side block rule. This is all the user will see after executing the malware. Nothing. They have no idea they were infected.



This is what happened to the infected host. Above I can verify the Zeus malware has fully installed on this system and is actively running without the user's knowledge or the company's consent.



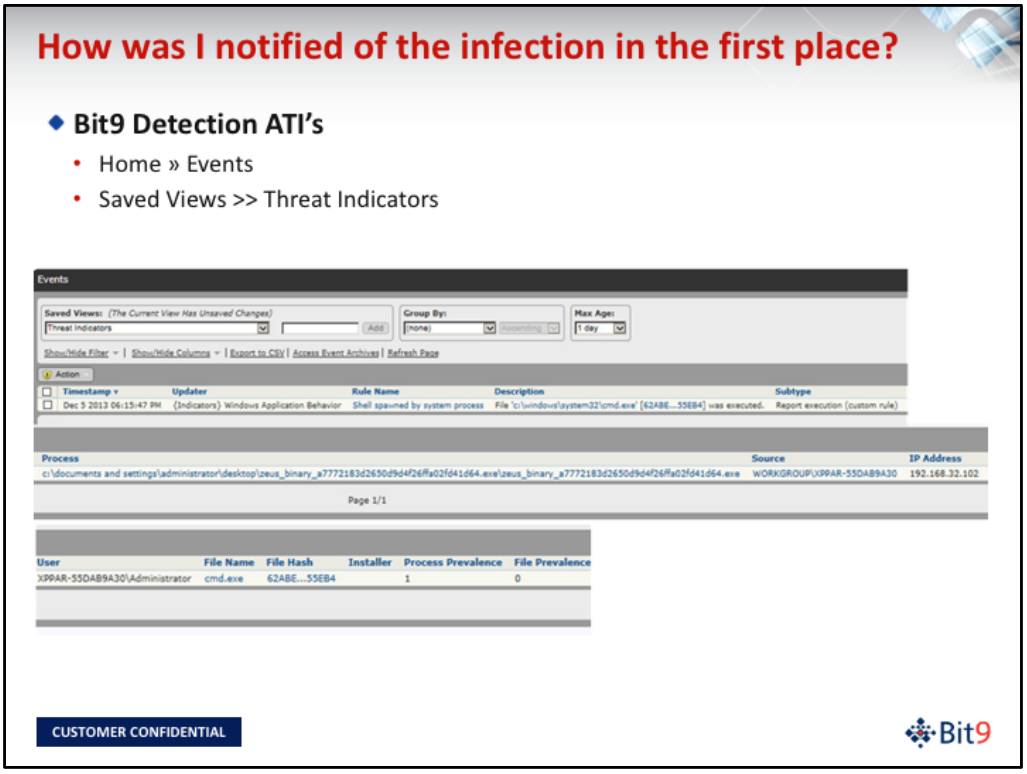
What happens if we were to approve the file in medium enforcement? The same thing that happens to the low enforcement machine without a block rule. The host becomes infected. This is a view of the infection from the console.



Now that we have a confirmed infected endpoint, how about we start this IR?

ask above questions

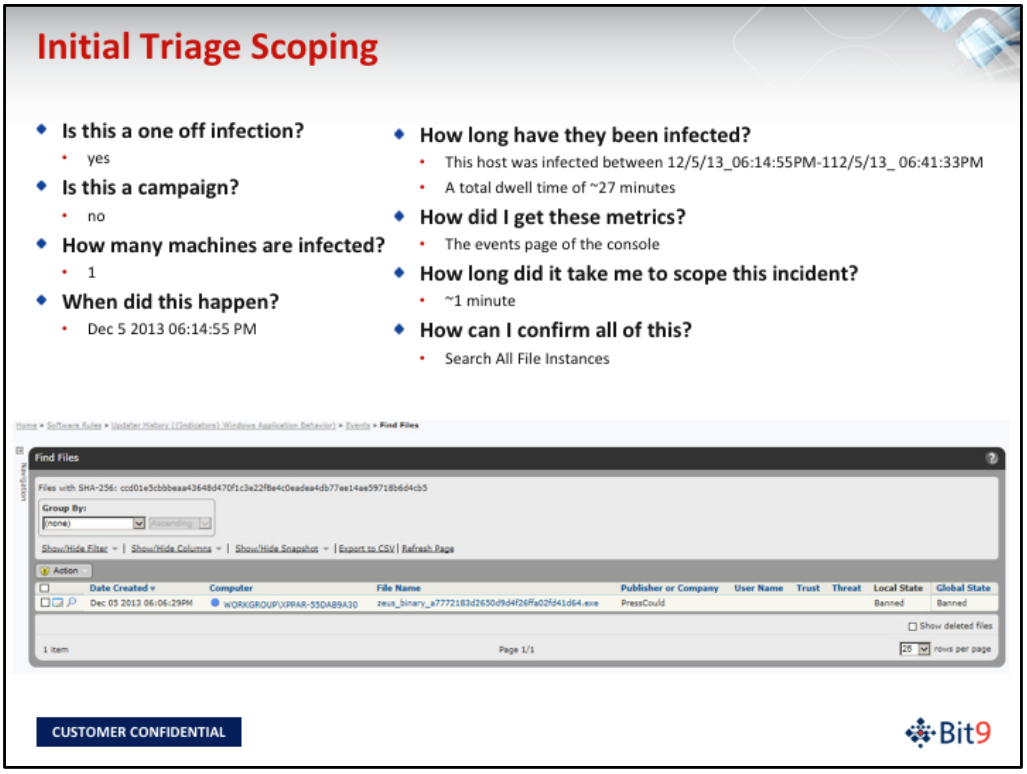
Let's answer a few of these really quickly using the bit9 console.



Bit9 Detection alerted us of the infection to the XP client. Zeus has a unique behavior that sets off our 'Shell Spawned by System Process" ATI.

After checking the "Threat Indicators" View in the console, I notice an alert for a file that I am unaware of. I should probably look into this event further.

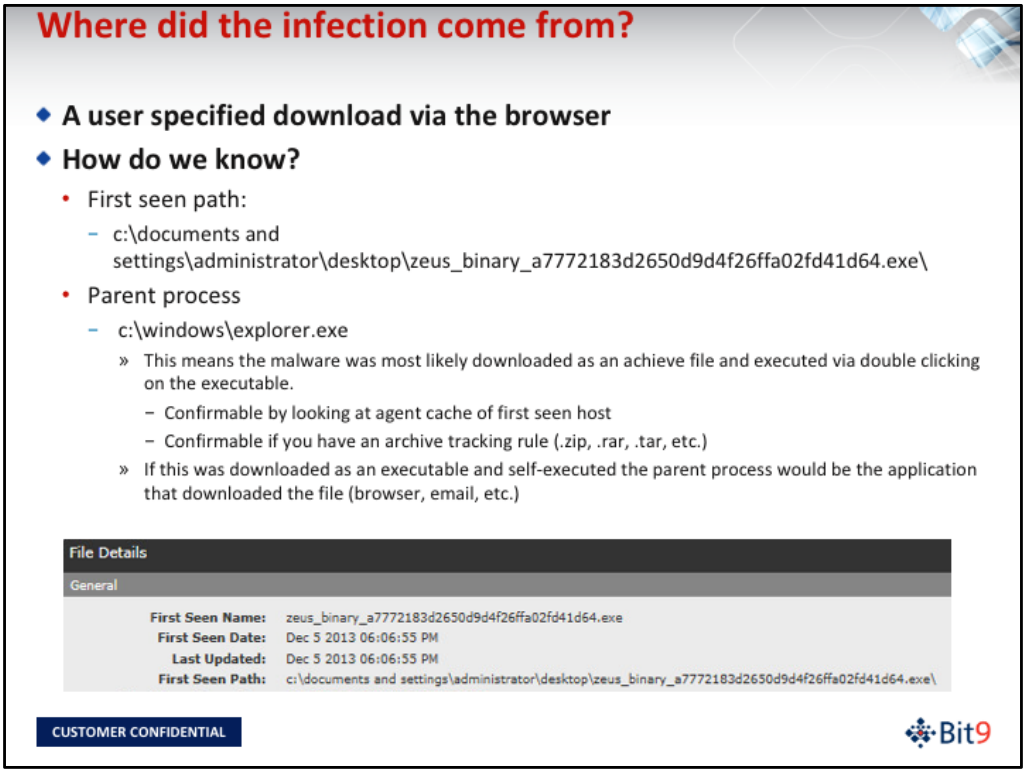
explain columns seen here



Now that we have been alerted to a suspicious behavior, lets answer some typical IR questions about this event for IR scoping.

go over above questions

We can see from the All File Instance page that this is a one off unique event and we should probably ban the file globally.

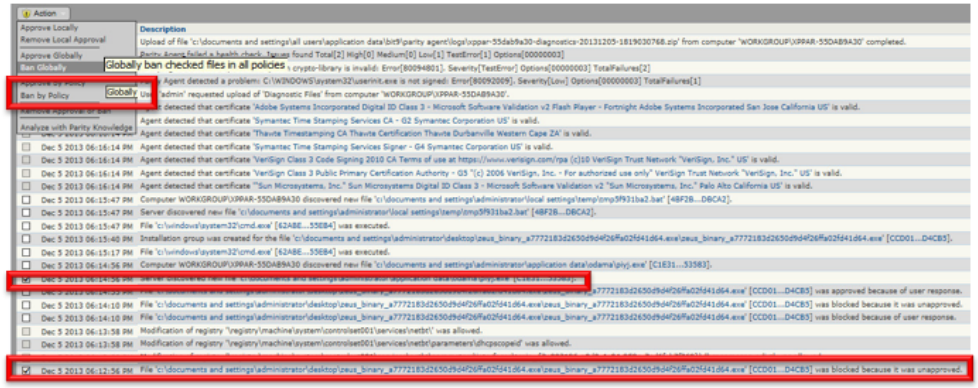


Before I show you how to ban this file, let's discuss where it came from

explain above

How do I stop the infection and spread of the malware?

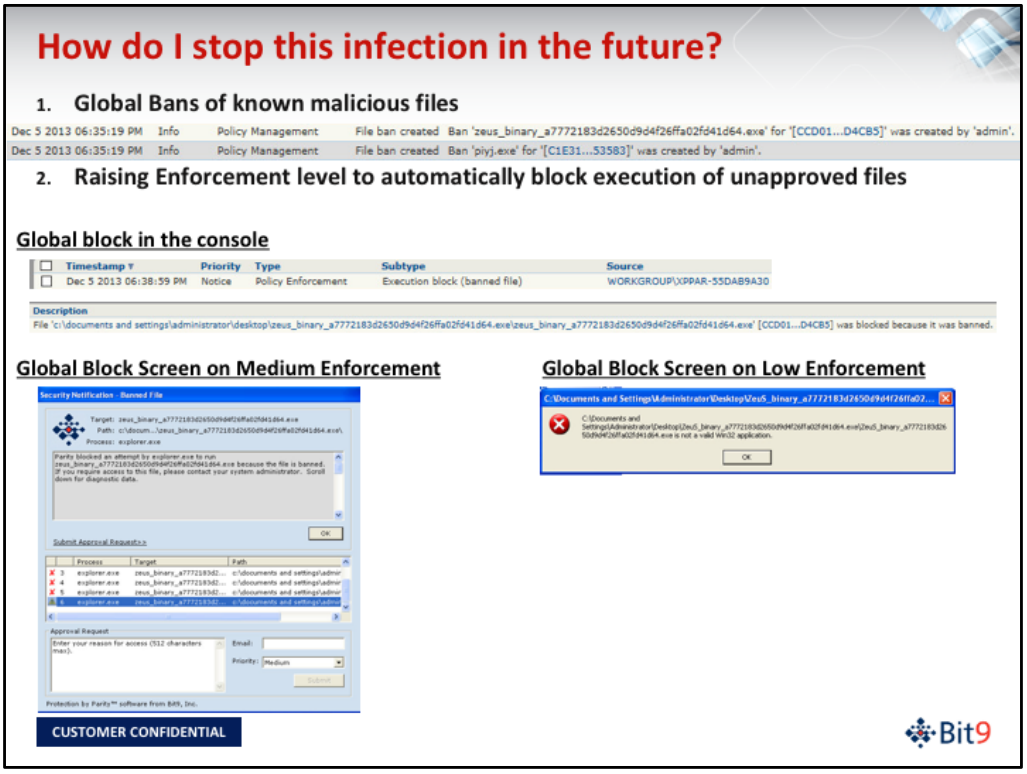
- **Global ban**
  - This is always my first step when finding unknown or suspicious files in my environment
  - Why am I not afraid of banning a good file?
    - Files are guilty until proven innocent
    - With Bit9 I can always easily and quickly un-ban a file



CUSTOMER CONFIDENTIAL

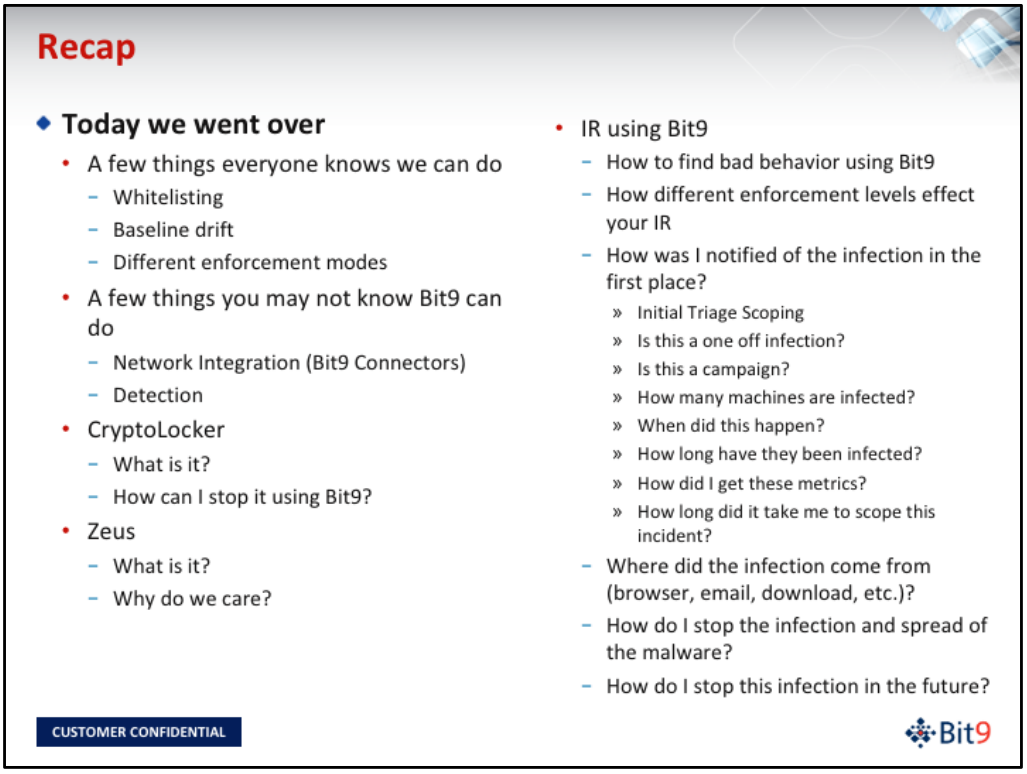
Bit9

The simplest and best solution here is to globally ban the file and all other executables associated with the file. Above is me doing this from the console.



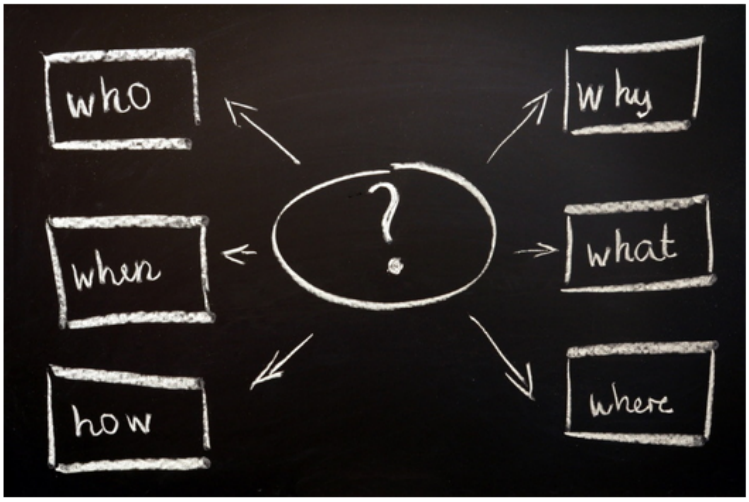
Now that we have this file banned, we can see from both the console, an endpoint in medium enforcement, and an endpoint in low enforcement that the malware cannot execute any longer.

explain above


Recap
Today we went over
A few things everyone knows we can do
Whitelisting
Baseline drift
Different enforcement modes
A few things you may not know Bit9 can do
Network Integration (Bit9 Connectors)
Detection
CryptoLocker
What is it?
How can I stop it using Bit9?
Zeus
What is it?
Why do we care?
IR using Bit9
How to find bad behavior using Bit9
How different enforcement levels effect your IR
How was I notified of the infection in the first place?
Initial Triage Scoping
Is this a one off infection?
Is this a campaign?
How many machines are infected?
When did this happen?
How long have they been infected?
How did I get these metrics?
How long did it take me to scope this incident?
Where did the infection come from (browser, email, download, etc.)?
How do I stop the infection and spread of the malware?
How do I stop this infection in the future?
CUSTOMER CONFIDENTIAL
Bit9

recap

Questions?



CUSTOMER CONFIDENTIAL

Bit9

questions