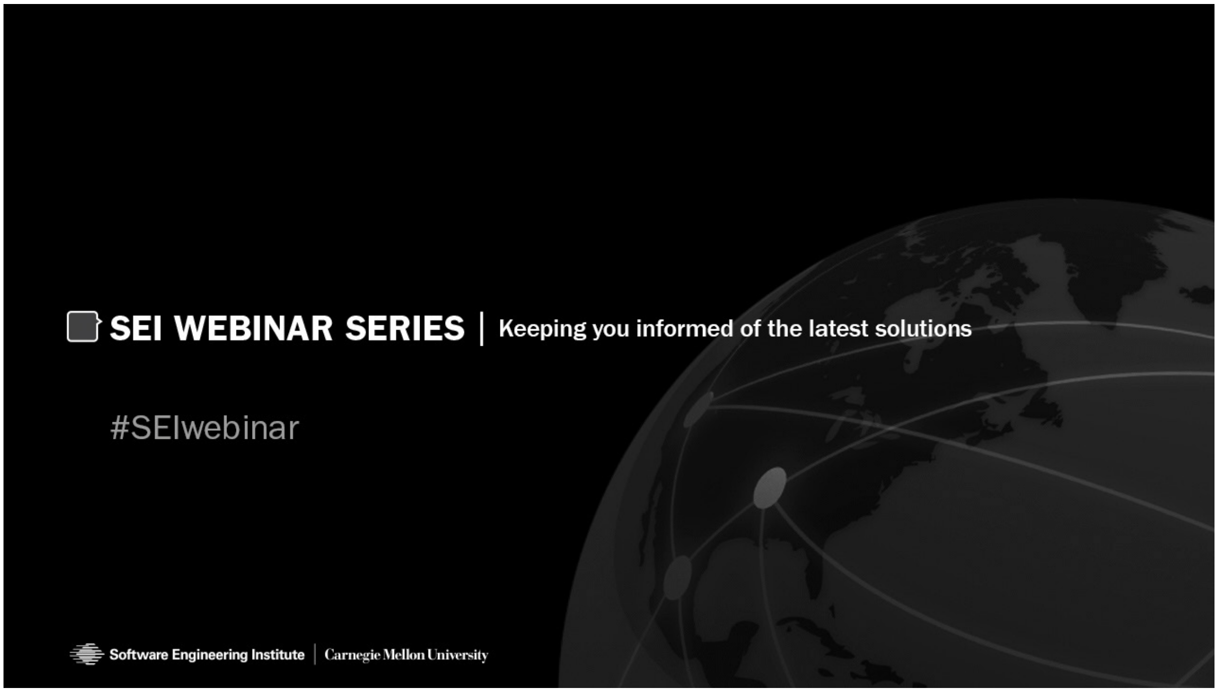
Building and Scaling a Malware Analysis System

Table of Contents

SEI WEBINAR SERIES | Keeping you informed of the latest solutions

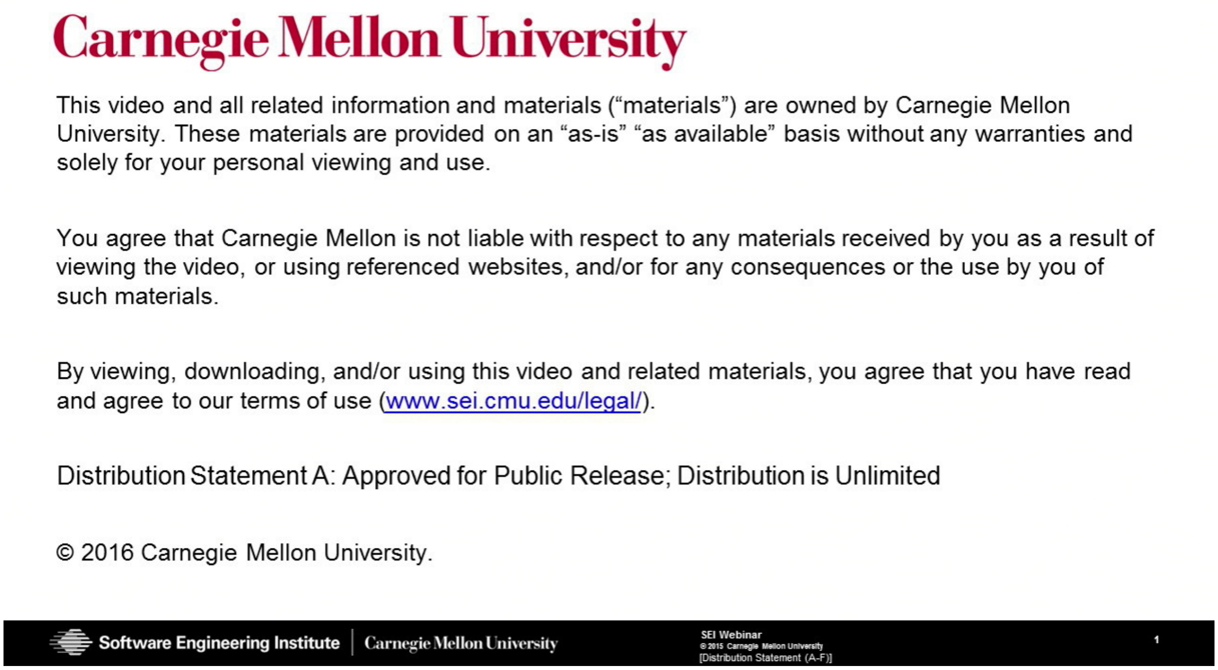


SEI WEBINAR SERIES | Keeping you informed of the latest solutions

#SEIwebinar

Software Engineering Institute | Carnegie Mellon University

Carnegie Mellon University



Carnegie Mellon University

This video and all related information and materials ("materials") are owned by Carnegie Mellon University. These materials are provided on an "as-is" "as available" basis without any warranties and solely for your personal viewing and use.

You agree that Carnegie Mellon is not liable with respect to any materials received by you as a result of viewing the video, or using referenced websites, and/or for any consequences or the use by you of such materials.

By viewing, downloading, and/or using this video and related materials, you agree that you have read and agree to our terms of use (www.sei.cmu.edu/legal/).

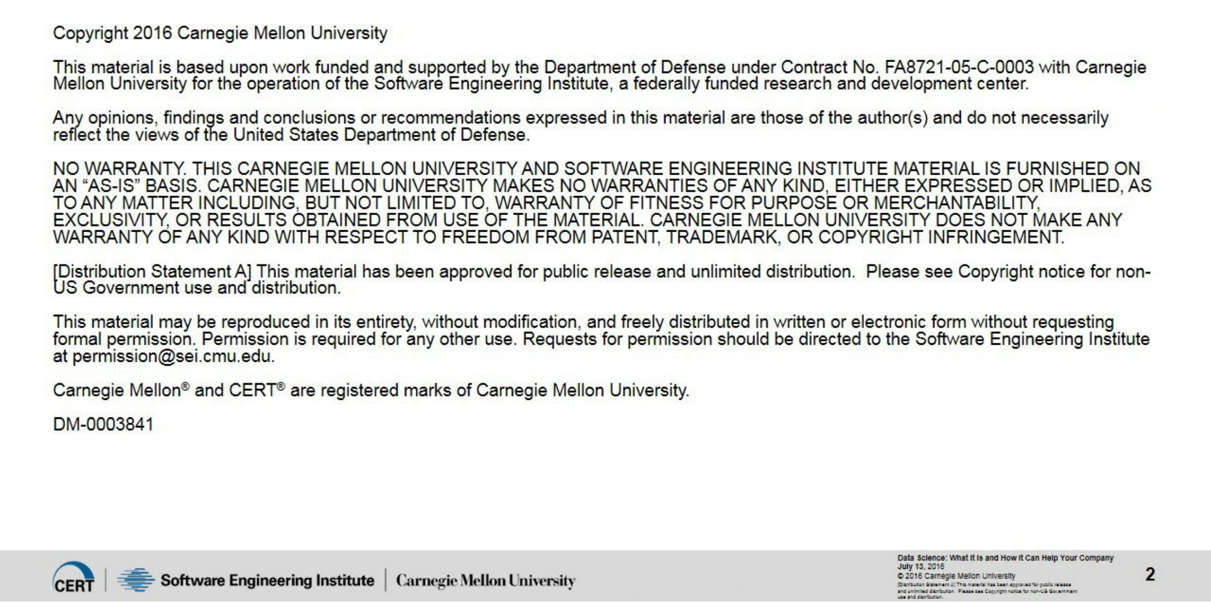
Distribution Statement A: Approved for Public Release; Distribution is Unlimited

Software Engineering Institute | Carnegie Mellon University

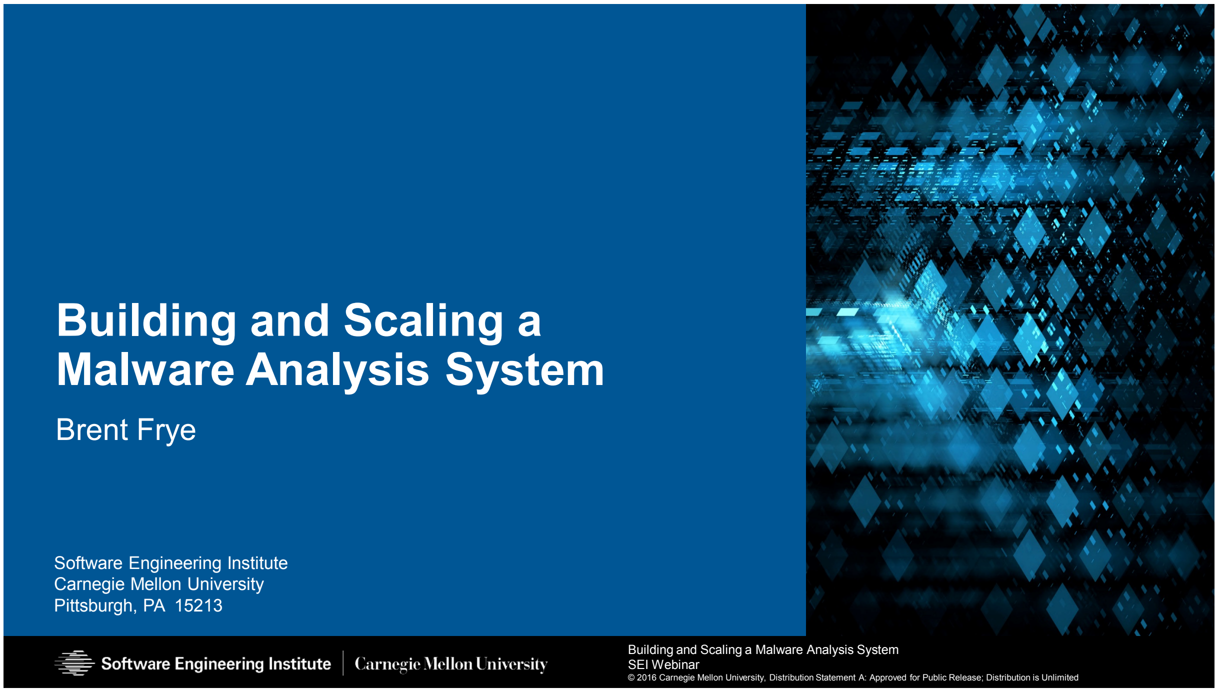
SEI Webinar
© 2015 Carnegie Mellon University
[Distribution Statement (A-F)]

1

Copyright 2016 Carnegie Mellon University



Building and Scaling a Malware Analysis System



Building and Scaling a Malware Analysis System

Brent Frye

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

**004 Presenter: And hello from the campus of Carnegie Mellon University in Pittsburgh, Pennsylvania. We welcome you to the Software Engineering Institute's webinar series. Depending on your location, we wish you a good morning, a good afternoon, or a good evening. Our presentation today is Building and Scaling a Malware Analysis System.

My name is Shane McGraw. I'll be your moderator for the presentation. And I'd like to thank you for attending. We want to make today as interactive as possible. So, we will address questions throughout the presentation and again at the end of the presentation. You can submit those questions to our event staff at any time through the chat or the questions tab on your control panel. We'll be monitoring both of those.

We will also ask a few polling questions throughout the presentation. And they will appear as a pop up window on your screen. The first question we'd like to ask is just how you heard about today's event. So, you can vote on that now.

Another three tabs I'd like to point out are the download materials, Twitter, and survey tabs. The download materials tab has a PDF copy of the presentation slides there now. The survey tab we ask that you fill out at the end of the event as your feedback is always greatly appreciated. For those of you using Twitter, be sure to follow cert_division and use the hashtag seiwebinar.

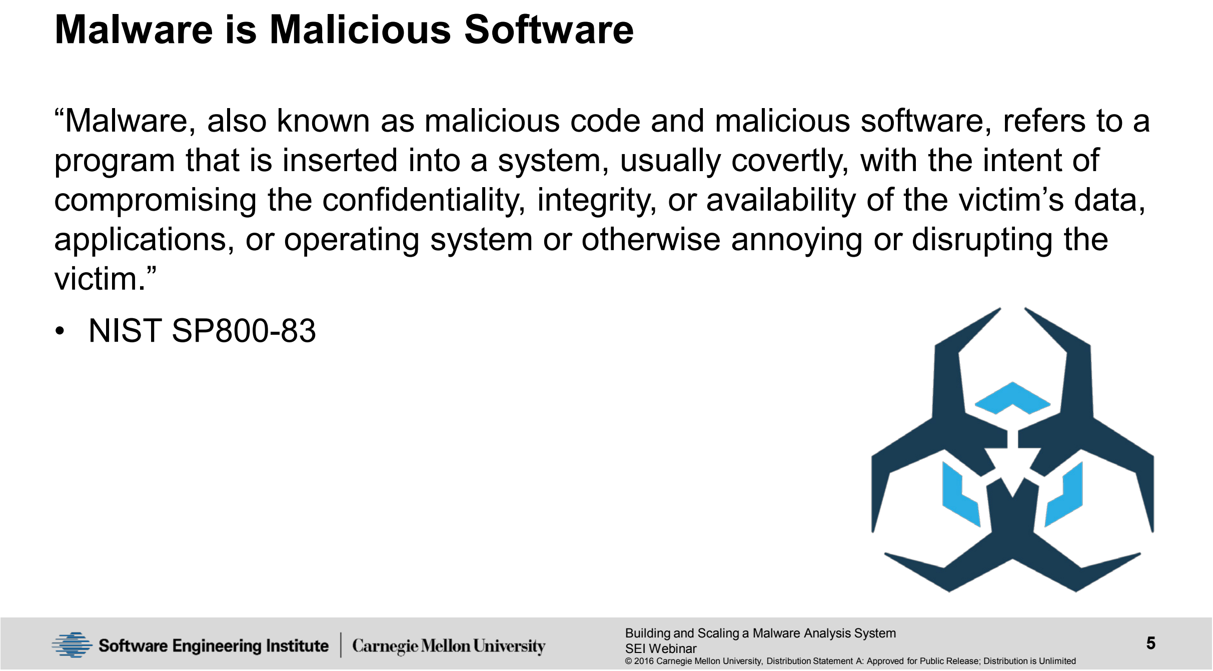
And now I'd like to introduce our presenter for today. Mr. Brent Frye is a member of the technical staff within CERT. His current interests include scaling, malware collection and analysis capabilities, process improvement, and data mining. He received his BS in computer science from Carnegie Mellon University. And prior to joining the SEI, Brent led a development team at a computer security aggregation and monitoring start up and served as a technical leader on teams at Carnegie Mellon and the University of Pittsburgh. So, now I'd like to turn it over to Brent. Brent, welcome. All yours.

Presenter: Thank you, Shane. So, to begin-- is it moving?

Presenter: Here, yeah.

Presenter: There we go.

Malware is Malicious Software



Malware is Malicious Software

“Malware, also known as malicious code and malicious software, refers to a program that is inserted into a system, usually covertly, with the intent of compromising the confidentiality, integrity, or availability of the victim’s data, applications, or operating system or otherwise annoying or disrupting the victim.”

- NIST SP800-83

Software Engineering Institute | Carnegie Mellon University

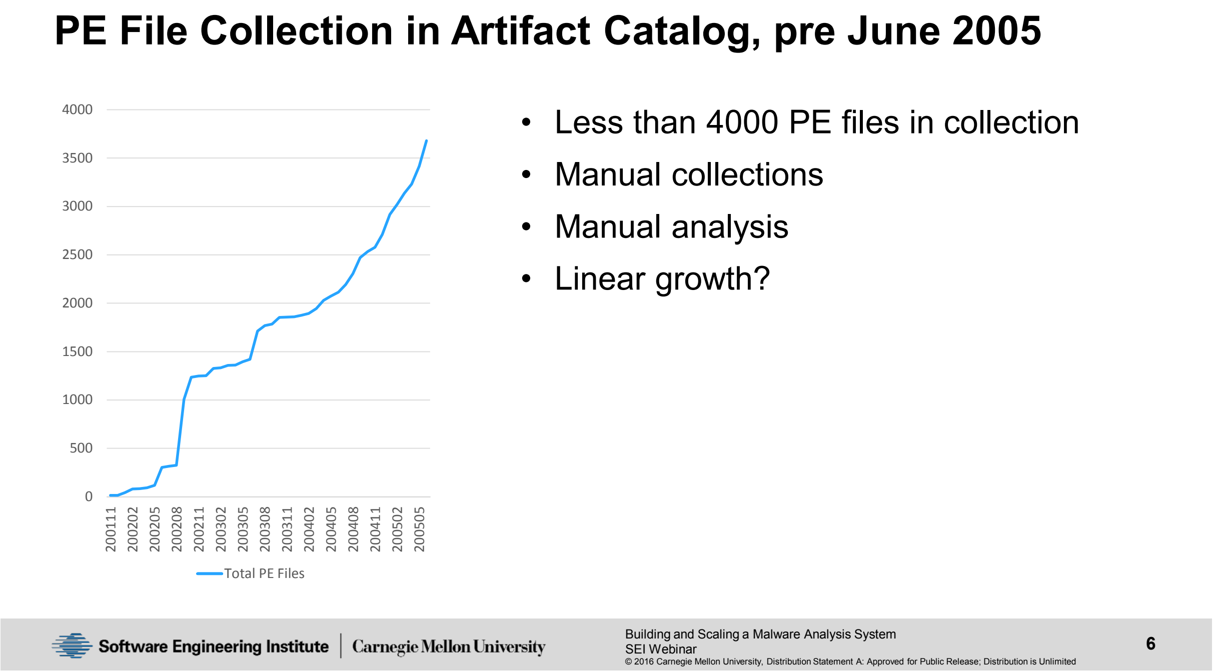
Building and Scaling a Malware Analysis System
SEI Webinar
© 2016 Carnegie Mellon University, Distribution Statement A: Approved for Public Release; Distribution is Unlimited

5

**005 Sorry about that. So, malware is malicious software. It's designed to overcome your computer's defenses and to make your computer do things that you don't know about and you probably wouldn't like it if you did. Most people who use computers have no need to find and analyze the malware that appears on their machines. They just want to protect against it and make sure that it doesn't remove any of their files or anything like that. And they just want to remove the malware if they find it.

But I suspect most of you watching this are somehow involved in learning more about what the malicious software actually does when it infects computers. I won't go into what to do to protect yourself while handling malware because that could be a whole webinar in and of itself. And it can be tricky to do that, especially considering most of the antivirus programs that people typically put on their machines will actively try to remove things when you are actually trying to analyze them.

PE File Collection in Artifact Catalog, pre June 2005


PE File Collection in Artifact Catalog, pre June 2005

- Less than 4000 PE files in collection
- Manual collections
- Manual analysis
- Linear growth?

Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar
© 2016 Carnegie Mellon University, Distribution Statement A: Approved for Public Release; Distribution is Unlimited

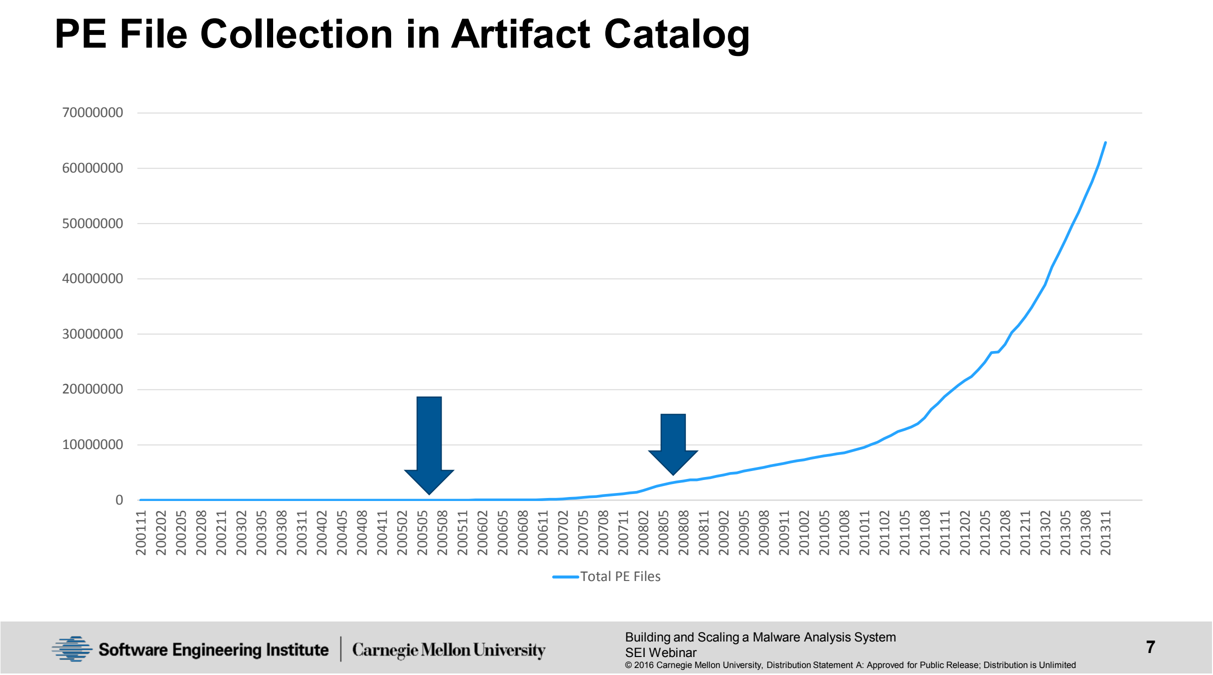
6

**006 So, when I started working at CERT in 2005, we had the artifact team that had been collecting data in our artifact catalog for about four years. We had a collection at that point of about thirty-seven hundred unique PE files, which are Windows executable files. And we were getting fewer than ten new PE files, unique PE files, every day. We had more files than that in the entire collection. But the focus at that point was in analyzing those Windows malware.

The collections process was mostly manual. When you're only doing ten a day, it really doesn't take a lot of effort to do that. But we were planning on scaling up that process. After the initial testing and data load, as you can see on this for the first little bit, the database started growing larger starting in late 2002. By the time I started, it was clear that we were going to continue to acquire more malware samples but not really how fast we were going to be collecting them. We didn't have enough data at that point. But based upon the previous figures, it looked like we were doubling our collection about every two years, with a spike if we got a sizeable collection from somewhere.

In hindsight, I can see on the second half of that graph, a little bit of a curve in the growth. But it could just as easily have just been noise. So, we weren't quite sure what to expect. And then we started automating our acquisitions process.

PE File Collection in Artifact Catalog



PE File Collection in Artifact Catalog

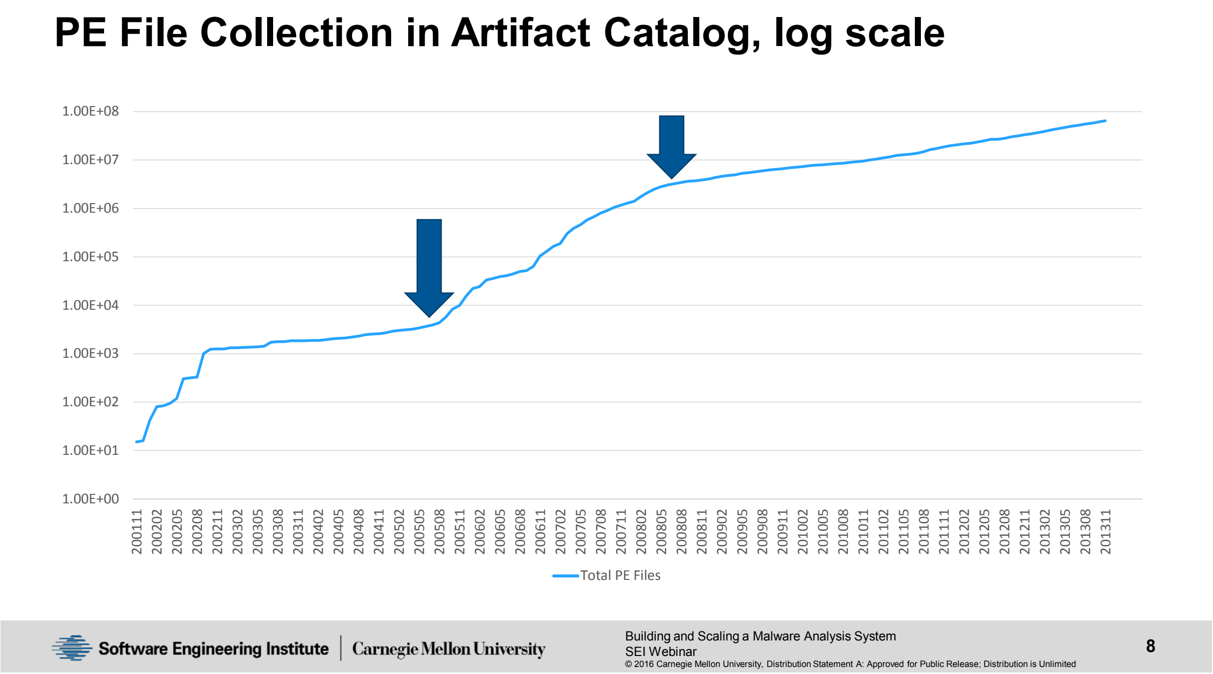
Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar
© 2016 Carnegie Mellon University, Distribution Statement A: Approved for Public Release; Distribution is Unlimited

7

**007 So, the big arrow on this graph is where we were on the previous slide. As you can see, it's pretty much noise compared to where we were eight years later. In fact, the first five years is just a blip there. By the end of 2013, we had acquired over sixty-four million PE files alone. The growth of that curve after the little arrow there was pretty steady through that whole time at about four point seven, four point eight percent per month, which means that we're doubling every fifteen months or so. All of the data in the catalog was growing at about that same rate through that whole time period as well.

PE File Collection in Artifact Catalog, log scale



PE File Collection in Artifact Catalog, log scale

Software Engineering Institute | Carnegie Mellon University

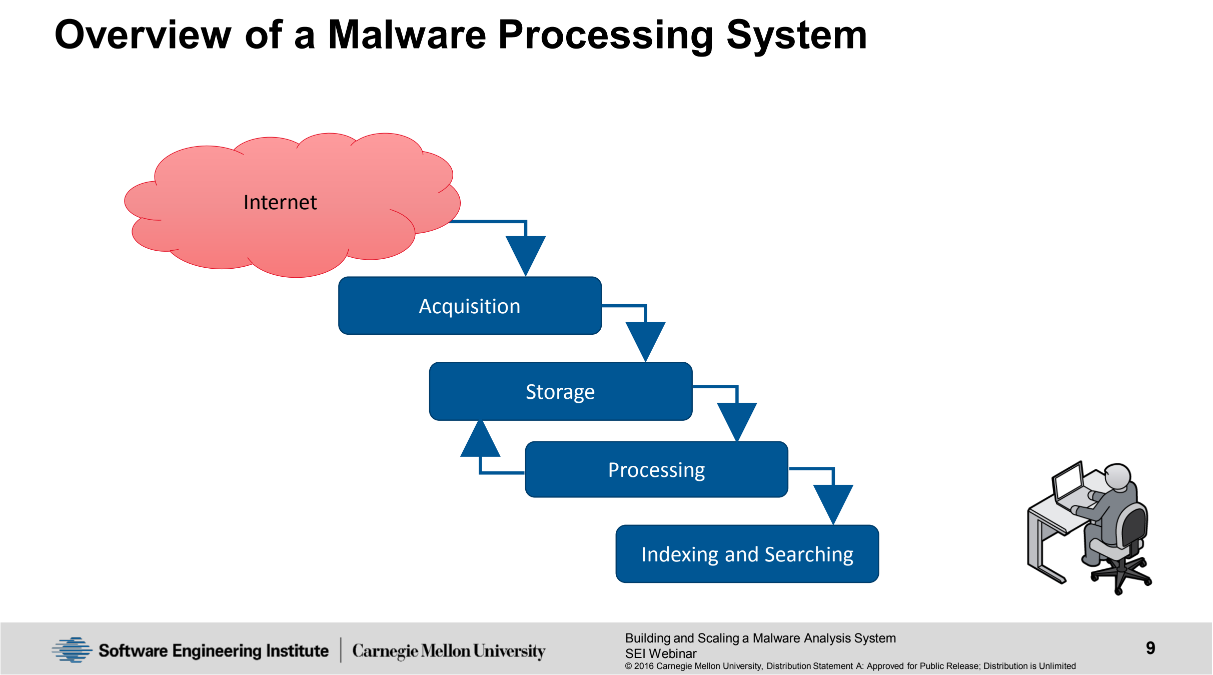
Building and Scaling a Malware Analysis System
SEI Webinar
© 2016 Carnegie Mellon University, Distribution Statement A: Approved for Public Release; Distribution is Unlimited

8

**008 This is the same data that was in the previous slide. But it's in a different scale. It's logarithmic. So, every step up is ten times more data. The nice thing about this kind of scale is that if you have an exponential growth, it shows up as a diagonal line. And the steeper the slope, the faster you're growing. And so, as I was saying, the line to the right of the small arrow there is at the four point seven, four point eight percent per month where we're doubling every fifteen months. To the left of the big line, it's about three percent where we're doubling every two years.

But what's really interesting for us at the time was the area in between where that represented a growth rate of about twenty percent per month where we were doubling every three point eight months, or about nine times the size every year. And so, by the end of that three year stretch in between those two, our catalog size had grown by almost nine hundred times its original size.

Overview of a Malware Processing System



Overview of a Malware Processing System

Internet

Acquisition

Storage

Processing

Indexing and Searching

Software Engineering Institute | Carnegie Mellon University

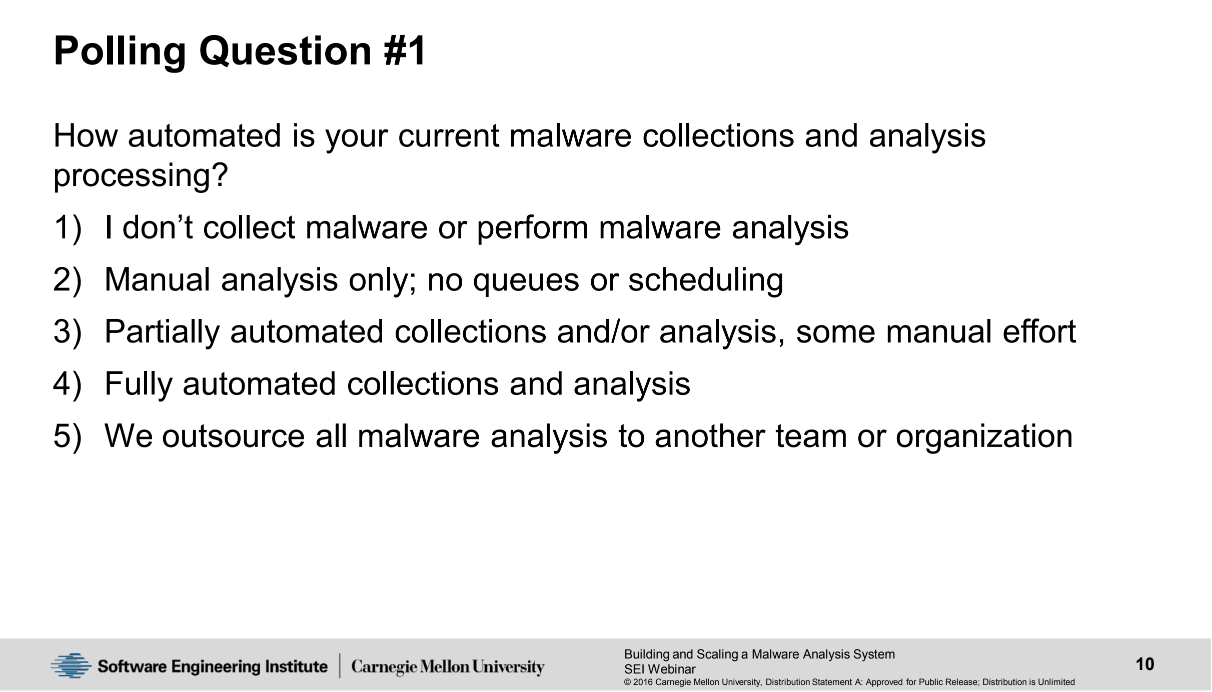
Building and Scaling a Malware Analysis System
SEI Webinar
© 2016 Carnegie Mellon University, Distribution Statement A: Approved for Public Release; Distribution is Unlimited

9

**009 Now, our artifact team wasn't just about collections. Acquiring malware was important to us primarily because we were tasked with analyzing malware and finding out why it did what it did. So, malware analysts reverse engineer the samples and run them through virtual machines to see what the malware does. We could reverse engineer them, and find out what makes it tick, and maybe find out who built the malware and how to protect against it.

In addition to automating our acquisitions process, we needed to automate our various processes to keep up with demand. We're getting nine hundred times more malware. We're not going to be able to use the same number of people to do that analysis. We have to automate something. So, at a high level overview, as we see here, we're going to acquire the files. We're going to store them somewhere. We're going to do some processing. After the processing, we'll put the data back into the storage system. And we're going to have some way of indexing and searching for our poor little user on the right to go ahead and try to find the things.

Polling Question #1



Polling Question #1

How automated is your current malware collections and analysis processing?

1) I don’t collect malware or perform malware analysis
2) Manual analysis only; no queues or scheduling
3) Partially automated collections and/or analysis, some manual effort
4) Fully automated collections and analysis
5) We outsource all malware analysis to another team or organization

Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar
© 2016 Carnegie Mellon University, Distribution Statement A: Approved for Public Release; Distribution is Unlimited

10

**010 Presenter: Okay, that's going to lead us to our second polling question, folks. It will appear on your screen now. And we would like to know, "How automated is your current malware collections and analysis processing?" So, we'll give you about ten or fifteen seconds to vote there. Brent, while we're waiting for that, we've got a couple questions coming in to the queue. One I think everybody's probably asking that may have no direct answer but more of a speculation, Joe wanted to know, "Why has the amount of malware grown so quickly over the years?"

Presenter: So, although in the news you see a lot about hacking attempts where places are getting hacked into and information stolen, that sort of thing, it's not necessarily the case that that's why we're getting more malware. It's not really-- I'm not really clear as to why we're getting more malware. It's-- or at least more malware types. It doesn't make sense that we would have millions upon millions of things, especially if we have malware families that have thousands or tens of thousands of members and are essentially doing the same thing. So, I don't know why there's been this big explosion. Although, there's certainly good reason for there to be malware continuing to be present in the computer environment.

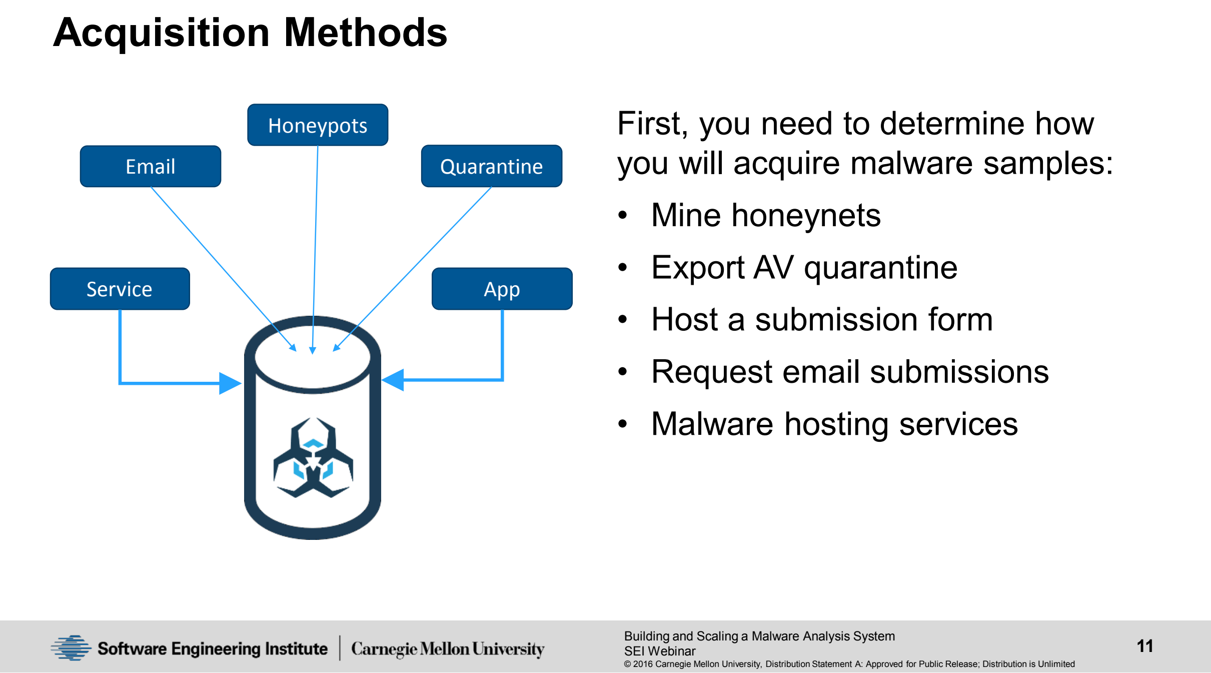
Presenter: Okay, and a question from Ed here asking, "Imagine a website with malware-containing advertisements. How would you acquire the malware from that website? Would it be an automatic or manual process?"

Presenter: I would imagine for something like that, you would want to automate some web browser or something that would refresh periodically to get new malicious advertisements. And then you would just load those one by one. If it's a high probability that they are going to be malicious, then you'd be keeping most of those. But you might just want to run them through like an AV system and say, "Is this malware or not?" But it also depends on what the technology is that's pushing those ads. It might be trying to prevent you from doing that. So, for a new site where you're doing that, you would probably want to have that be a manual process until you understand what the mechanics are so you could automate that better.

Presenter: Okay, great. Just real quickly to go back to our polling question, we had thirty-nine percent I don't collect malware or perform malware analysis. So, that was thirty-nine percent. Twenty-four percent manual analysis only, twenty-seven percent partially automated, two percent fully automated, and seven percent we outsource all malware analysis. And I'll push those to the audience.

Presenter: All right, well that's an interesting combination of that. I wasn't sure how many of the audience would actually be interested in malware or had actually done anything with it. So, it looks like a good proportion is actually interested in malware analysis. And about a third of you have had some attempts to do automation there. All right, so that will help me going forward here. It's back to this.

Acquisition Methods



Acquisition Methods

First, you need to determine how you will acquire malware samples:

- Mine honeynets
- Export AV quarantine
- Host a submission form
- Request email submissions
- Malware hosting services

Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar
© 2016 Carnegie Mellon University, Distribution Statement A: Approved for Public Release; Distribution is Unlimited

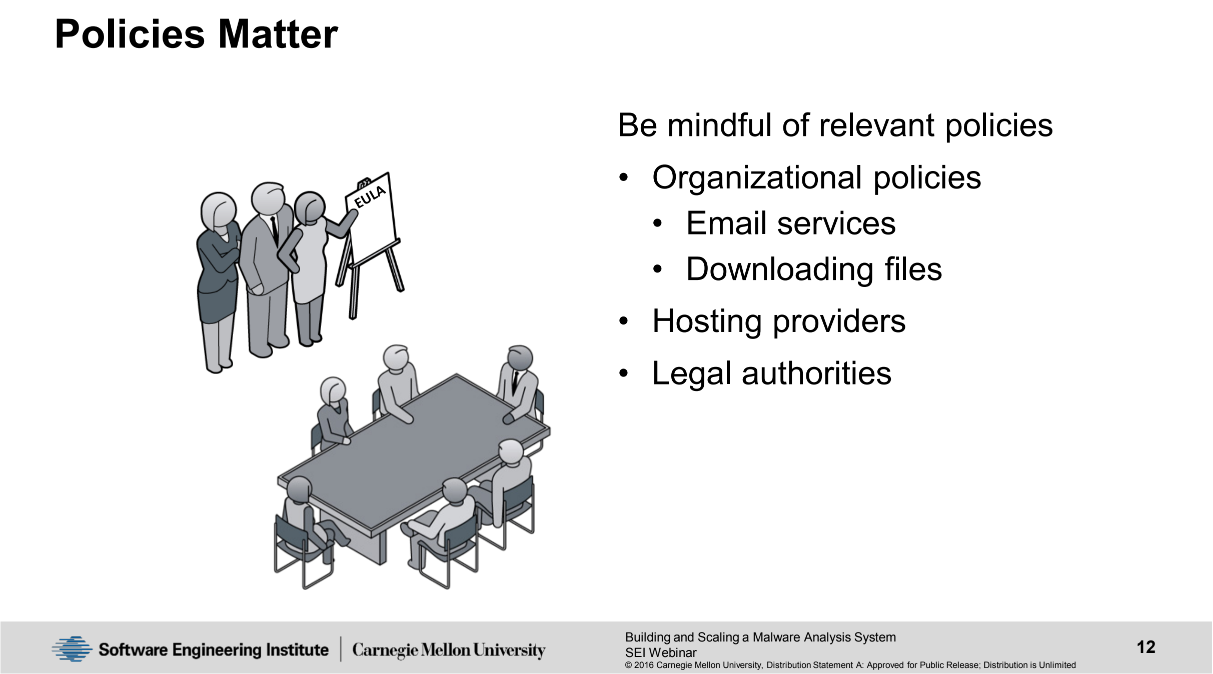
11

**011 So, acquisition, so I was saying that we automated the acquisition process. And we need to acquire the malware from somewhere. And we've got a number of possibilities, although some of them are low yield and probably wouldn't be good for a large scale network analysis system. You could try and set up a honey pot or a honey net and mind that and collect data from there.

You could go to your IT shop. Most IT shops, most enterprises nowadays will have an AV system that looks at all their incoming malware-- or, sorry incoming files and determine whether it's a malware or a normal email. And if it determines that it's malware, it will put it in a quarantine. And you could possibly ask your IT shop to deliver you the things that got quarantined.

You could host a submission form or request email submissions if it's something that's going to work along with your core business. The app that's on here, things like antivirus or operating systems, if it determines that something that's malicious is going on, it might prompt you or have a check box that you sign in when you install the software that says submit this to the server if we find something. And so, if you're one of these kind of companies you can collect malware that way. But that's not for the masses. If you're trying to do research, you could try to do a malware hosting service, either a legitimate one, which is providing samples for research purposes, or an illegitimate one, which would be somebody who has malware set up for people to use for their own malicious purposes.

Policies Matter



Policies Matter

Be mindful of relevant policies

- Organizational policies
  - Email services
  - Downloading files
- Hosting providers
- Legal authorities

Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar

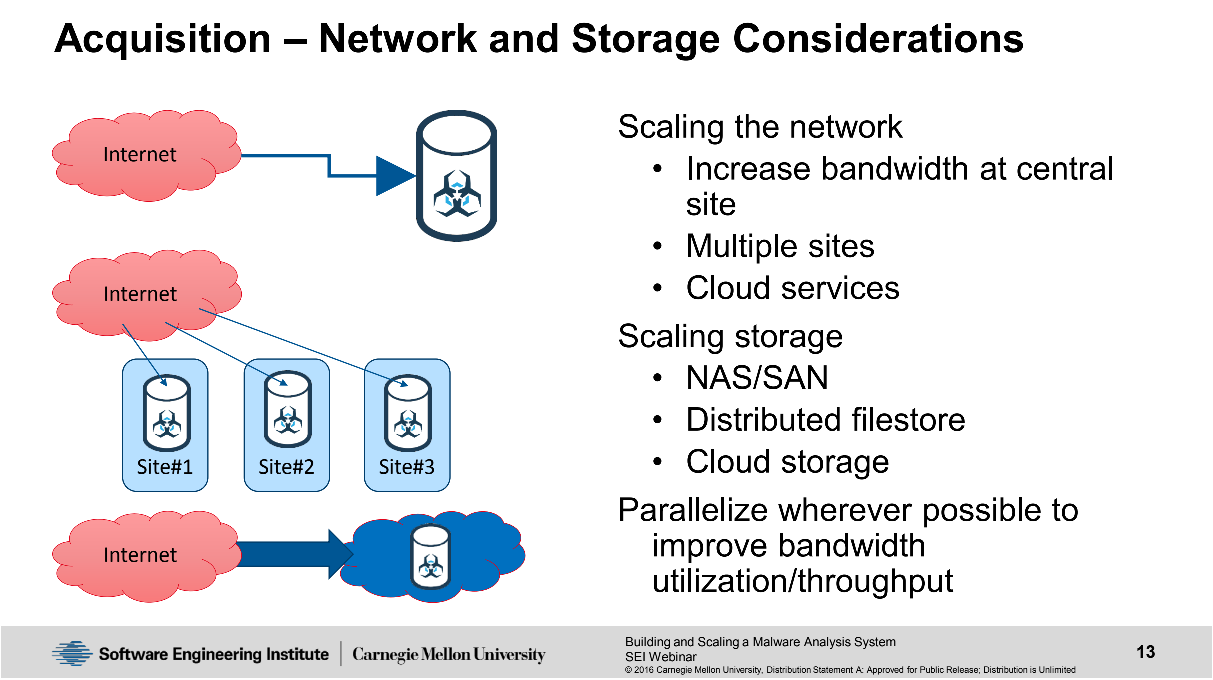
12

**012 When you're doing this in an enterprise environment, you need to be aware of what your organizational policies are because oftentimes, you are limited as to what can or cannot be on your networks and services. I already mentioned enterprise email services are often being protected by antivirus products. And so, web access might also be restricted. And so, if you're going to be using either of these two things to acquire your malware, you need to make arrangements so that it's not blocked and you won't be getting anything.

You need to make sure that you have appropriate permissions. The last thing that any company wants is to be in the news because somebody at their site let a bunch of malware loose and started infecting people. You want to let your IT department know so that they won't shut you down if you do something that is normally prohibited by policy. And you want to make sure that your service providers won't disconnect you when they notice that malware traffic is coming on their networks, through their networks, or on their disks depending on the provider.

And you want to make sure that analyzing malware is legal in your jurisdiction. I understand that many of the audience might be from outside the United States. And while this processing might be legal in the United States for research purposes, they may not be legal where you are. And so, you need to get advice on that.

Acquisition – Network and Storage Considerations



Acquisition – Network and Storage Considerations

Scaling the network
- Increase bandwidth at central site
- Multiple sites
- Cloud services

Scaling storage
- NAS/SAN
- Distributed filestore
- Cloud storage

Parallelize wherever possible to improve bandwidth utilization/throughput

Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar
© 2016 Carnegie Mellon University, Distribution Statement A: Approved for Public Release; Distribution is Unlimited

13

**013 The line between acquisition and storage seems pretty clear until you actually try and write a slide about it. So, the coder in me wants to merge these two into a single process. And while I can mostly resist this, it seems like here's a good place to talk about both because there are some choices that affect both. Whether your expected growth rate is five or twenty percent per month, you can only acquire the files that get to you. And you can only process the files that you can store in some way.

There's three basic mechanisms I have here that are available to do that. You can-- at the top, you can simply add more networking capacity to a central location. More bandwidth will probably cost you more if you get it from the same vendor. And in some cases, there might be hard limits to the amount of network bandwidth that's available to a given location. It's simply a matter of what's available on your ISP.

If you have geographically dispersed locations, you can try splitting up the acquisition effort among your different locations. Obviously, managing multiple data centers and network connections will mean more administrative overhead than a single site. But if you can't get more network to one site, adding bandwidth to multiple sites might just be your best option.

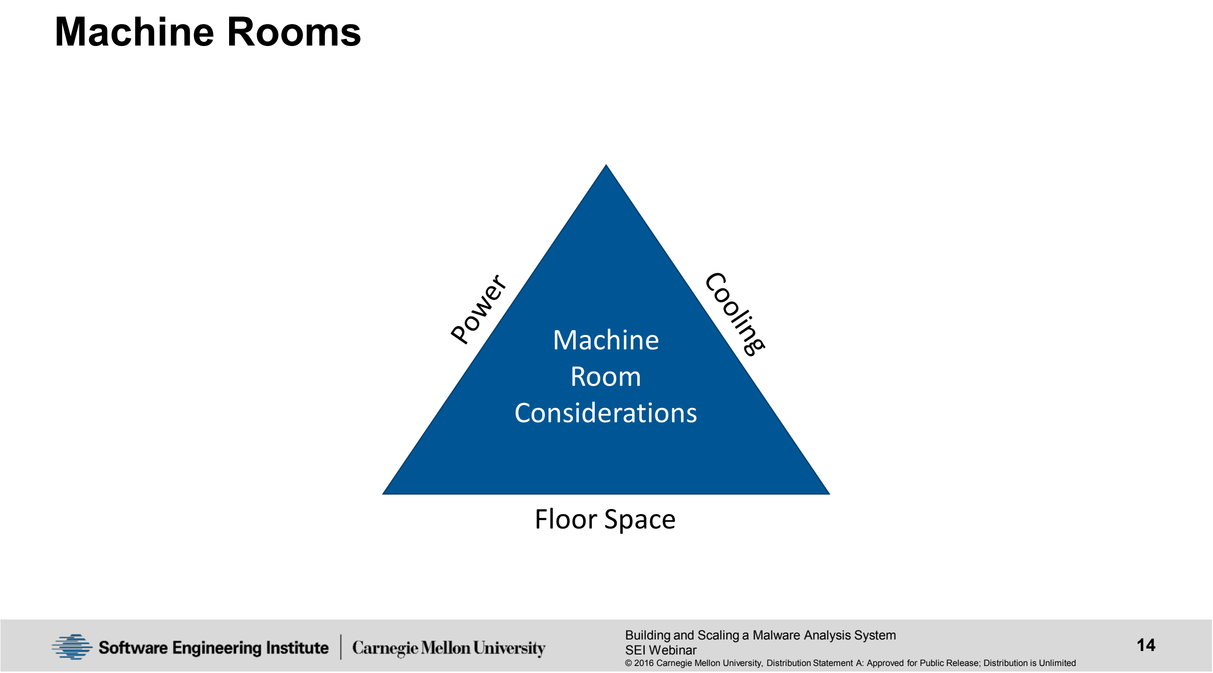
Or, you could try using a cloud-based system to acquire and store malware samples. This takes the network overhead away from your facility. And it may be the only way to get everything you want into one place at whatever data rate you're expecting. This option also makes sense if the intended storage location is also the cloud-based system where you are planning on doing the analysis.

If you don't plan to use cloud-based storage, you can build a large scale storage system with a variety of NAS or SAN products. Many of these are scalable or at least extensible. But they generally aren't built for the lots of small files system that we get when we're dealing with malware. And there's often a performance penalty once they start getting full. And as you start extending them, at least with more disks, unless you also add more head nodes or whatever they're called for that system, you're going to get slower throughput as you get larger and larger.

One potential advantage to a centralized SAN or NAS solution is that it can be accessed directly by the analysts if you want it to. So, the users of the system can run whatever tools they want to cover any of the files if they want to. And that can be a great benefit for the analyst as far as an ease of use concern. But the disadvantage is pretty much exactly the same thing. If you have a large number of analysts all trying to get to the entire system and looking at files with whatever tools they want to, you're going to start bogging down the system and get poor performance and a bunch of unhappy analysts.

So, there are a few distributed data store options available, HDFS and S3 are probably among the best known. One advantage to like an HDFS system is that the data storage nodes are also compute nodes. And you can-- by increasing storage, you'll keep your processing time relatively consistent and possibly be able to improve over time.

Machine Rooms



Machine Rooms

Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar
© 2016 Carnegie Mellon University, Distribution Statement A: Approved for Public Release; Distribution is Unlimited

14

**014 If you're not using cloud storage, then you need to worry about your server room constraints. Your storage system might be expandable. And the distributed hosts that you have locally might allow you to horizontally scale indefinitely. But the machine rooms and data centers have limits in terms of the square footage available, the power distribution, and the cooling. You can only double your collection size so many times before you run out of one or more of these three things.

I've heard a story about someone wanting to purchase a SAN system that was about half a petabyte, five hundred terabytes of data in a single rack, which sounded really cool until they talked to their IT folks who said that while they had the power, and they certainly had the floor space, they didn't have sufficient cooling capacity for that system. And luckily that was discovered before they actually purchased the system and installed it. So, they were able to figure out what it needed to be. And they could split it up in to two half-rack systems where the density wasn't so bad. And they could actually work with the cooling system that they had available.

Presenter: Would there be a reason there why they wouldn't do a cloud solution for there? Is it something that's security related that they would stay away from that?

Presenter: There might be a possibility with security if-- so, the problem with cloud systems is that it's somebody else's system.

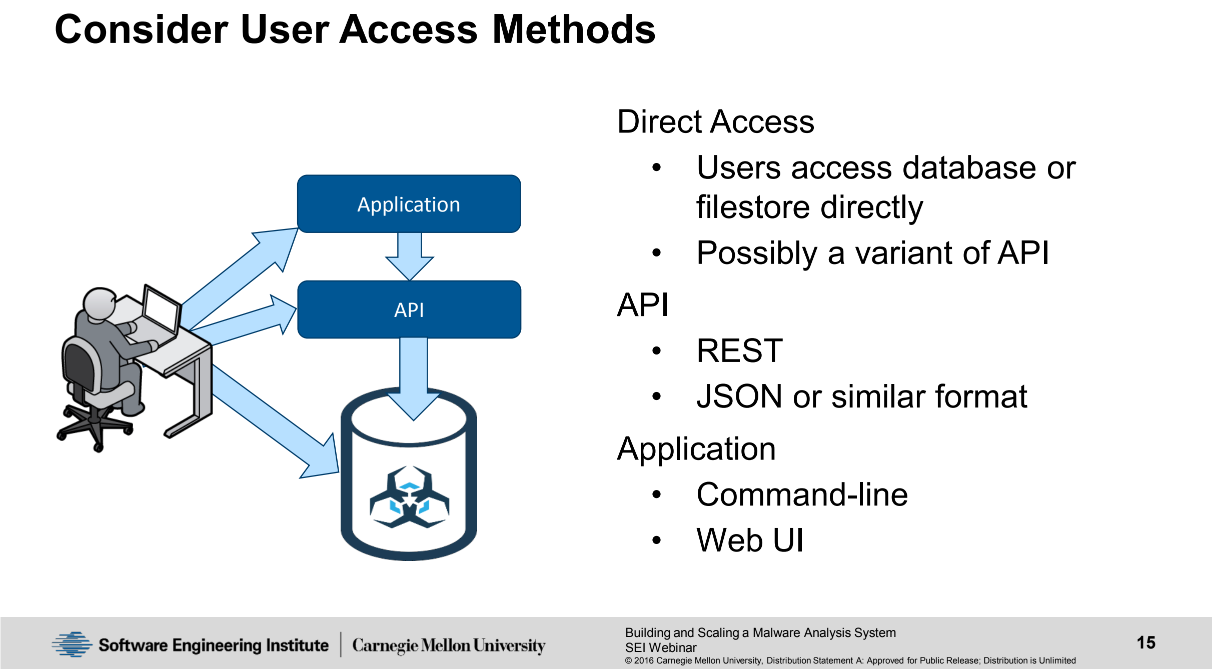
Presenter: Right.

Presenter: And you never really know what is going to happen on that other end. And if you have-- for the cloud storage, if you're not also doing the analysis there, that could also increase the latency and make it longer analysis timing as you're moving files back and forth.

Presenter: Okay.

Presenter: All right.

Consider User Access Methods



Consider User Access Methods

Direct Access

- Users access database or filestore directly
- Possibly a variant of API

API

- REST
- JSON or similar format

Application

- Command-line
- Web UI

Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar
© 2016 Carnegie Mellon University, Distribution Statement A: Approved for Public Release; Distribution is Unlimited

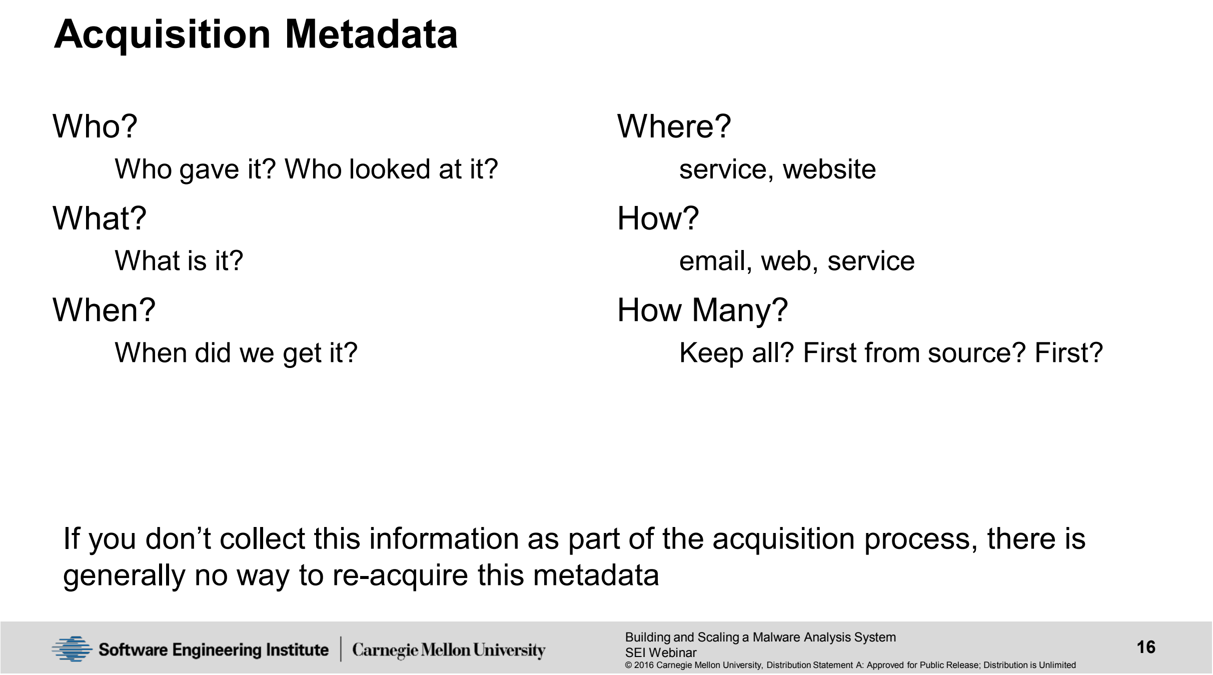
15

**015 So, before considering or committing to a storage solution, you want to consider how the users are eventually going to access the files and data. And I already mentioned one of the possible advantages of a NAS or a SAN system. But the issue I'm getting at here is how the users are going to get to other parts of that system as well. So, if they have direct access to the files or the database, that can actually limit the flexibility of your back end systems. You also want to keep in mind that while some users might just want a single file, others might want thousands or hundreds of thousands of files. And they might not be grouped together in whatever grouping scheme you're using.

But if you take direct access away, then you can actually do more things to improve the repository as far as making it more convenient to administrate. So, if you still provide files on demand with flow latency, in the back end, you can compress the files, or merge them, or split them up, or arrange them in to some other type metric that you think will improve performance. If you provide something like a rest interface that would allow users to access their files programmatically to get them out of the system or to control some other process that will perform the desired tasks over the specified files without needing to move the files to the user, then that can be a benefit there.

If the underlying software architecture is changed, and you have an API that's hiding that from you, then you can make changes to the system and make that transition seamless to the users. You could also provide a one stop shop with a command line tool set or a web user interface. That could provide a simple front end to whatever API you have. Or it could add a whole lot of additional features if you want it to.

Acquisition Metadata



Acquisition Metadata

Who?

- Who gave it? Who looked at it?

What?

- What is it?

When?

- When did we get it?

Where?

- service, website

How?

- email, web, service

How Many?

- Keep all? First from source? First?

If you don't collect this information as part of the acquisition process, there is generally no way to re-acquire this metadata

Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar
© 2016 Carnegie Mellon University, Distribution Statement A: Approved for Public Release; Distribution is Unlimited

16

**016 Presenter: So, Brent just while we're still in the area of cloud, a question from Derek came in saying, "Have you found any particular cloud service providers to be more or less apprehensive or willing to store known malware? Or, is the actual content a non-issue?"

Presenter: We have been looking at cloud providers, but to the best of my knowledge, we haven't made any inroads into actually finding out who would be a good solution for that. And so, I don't have an answer for that question

Presenter: And that's fine. I imagine it would be something that would have to be known up front because of the risks.

Presenter: But yeah, it's definitely-- you definitely want to be up front when you're discussing this with the vendors. Okay, this is what I'm going to have.

Presenter: Right.

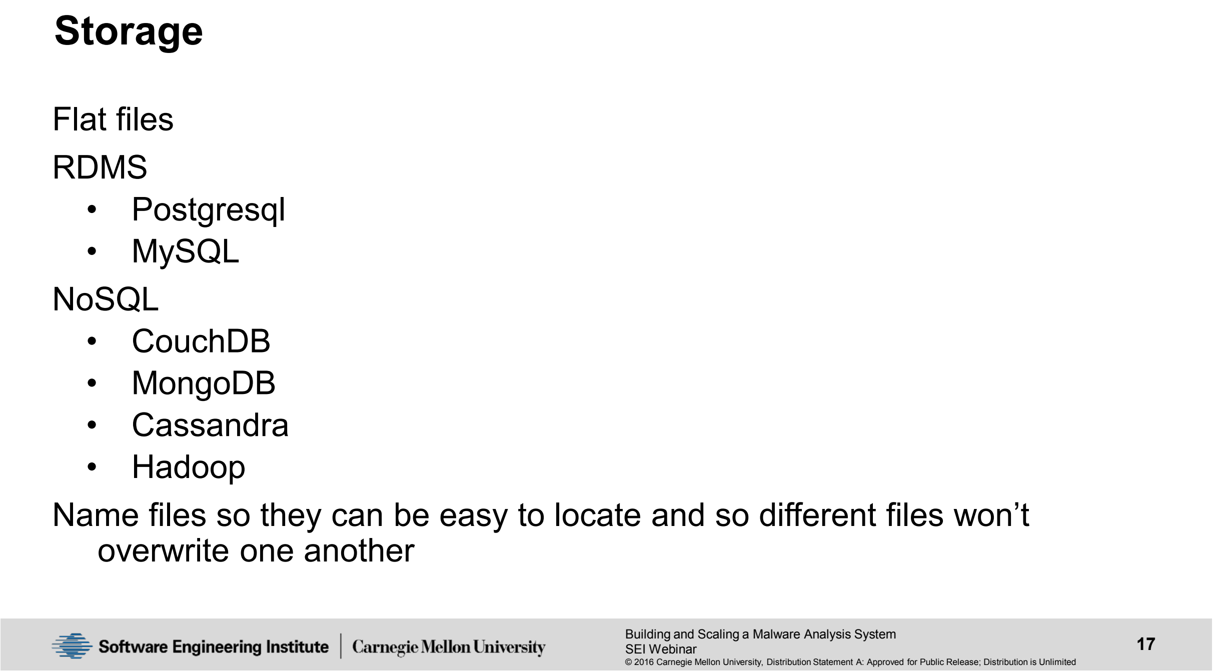
Presenter: All right, back to the slides. So, with-- we've talked about getting the files. So, whenever we have the files, we're going to, at this point, start collecting data. And we have available to us the, what I'm calling here, acquisition metadata. I haven't heard it referred to as anything else, or as anything actually. But for some uses, we would want to save information about where we got it from.

So, this category is anything that doesn't have to do with the malware sample itself. No matter how much we looked at the malware sample, we wouldn't be able to find any of this information of who, what, where, when, how, or how many. It's all up to whatever it is that we decide to do to retain whenever we put the malware in our system.

Most of these are self-explanatory. The one big question that you need to consider is the how many question. Whenever you are storing the data on the file system, you're probably just going-- no matter how many times you get it from a source, you're only going to store it once. And it will only take up that amount of disk space.

But you will want to possibly limit the metadata as far as who gave it to you. Maybe you're only interested in the very first time we got the data. Or maybe we want to know every single time anybody gave it to us no matter if they've given it to us twice or two thousand times. We want to know every single time they give it to us. My personal preference is wanting to know the first time we got it from a particular source. If we get it a second or third time, and it's just the same data, and it's the same file from the exact same person or place, that doesn't interest me in what I'm doing. It might interest some other people though if what they're looking for is information about the prevalence of metadata-- or the prevalence of malware, I mean.

Storage



Storage

Flat files

RDMS

- Postgresql
- MySQL

NoSQL

- CouchDB
- MongoDB
- Cassandra
- Hadoop

Name files so they can be easy to locate and so different files won't overwrite one another

**017 And the right answer to that

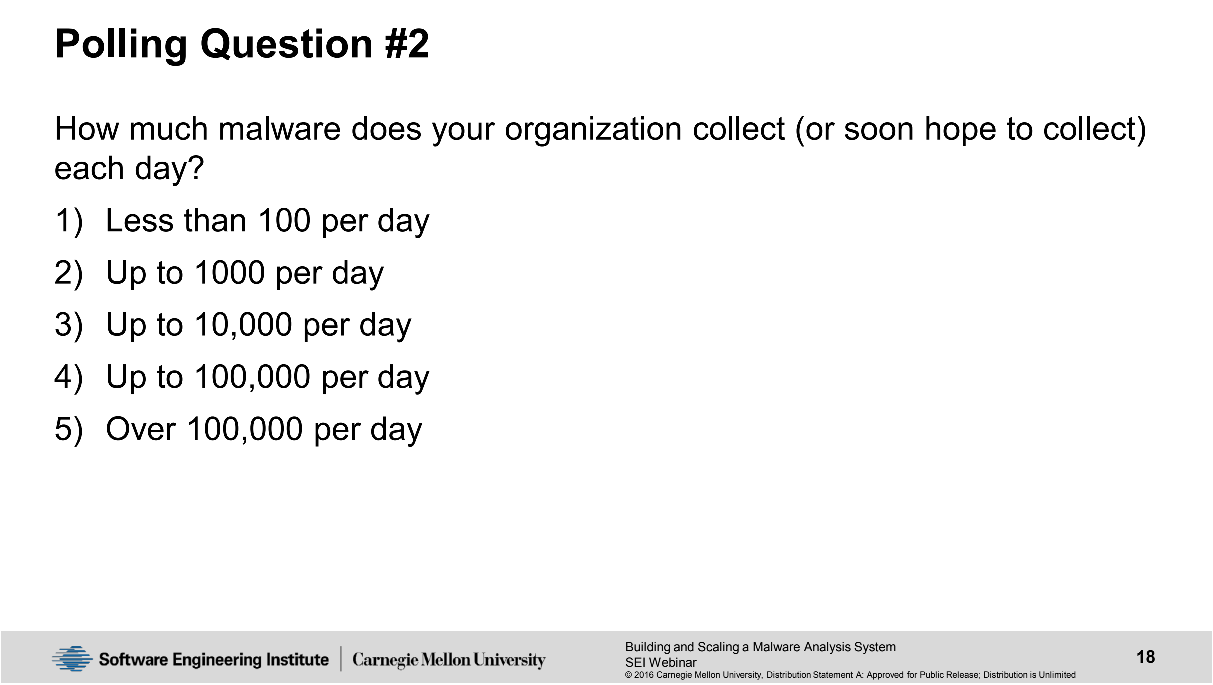
is really, like I said, what you want to do with the data. Now, I've already talked about storage some. But I was focusing there on storing files. And there's also the question about how to store the information that we collect about the files in way that users can access whatever the results are of our processing. And flat files can be useful in a number of situations where each bit of information is stored as one or more files.

When you have a lot of files to process, it's unlikely that file locking will be an issue, especially if the processing steps are done in a sequences so that only one agent is working with the file at any given time. And while where getting information out of a known file is easy with this particular system where you're dealing with just flat files, it can be a problem to try and find data later on once you get hundreds of millions of files or even hundreds of thousands of files.

So, a traditional relational database system is a pretty reasonable approach for much of the metadata that we want to store. The two I have mentioned here will easily handle hundreds of millions of records, hundreds of millions of rows for each file or for all the files combined. But it does have a disadvantage in that they typically don't deal with unstructured data well. And by unstructured data, I don't necessarily mean it's chaotic. I mean that the data itself individually might be very structured. And we'll see some of that when we get to the dynamic processing later on. But there might be too many variations of things that we'll be getting. So, you might need a table for anything that's dealing with file rights that is completely separate from network transactions and that is separate from UTXs and that sort of thing. So, the relational database approach is good if you know exactly what you're getting ahead of time. And you can build tables around that.

The no SQL solutions that I mentioned here are typically designed to be better at handling the unstructured data where the different rows that are in there for each data point, they'll have labels for them. But you don't need to know ahead of time what those labels are. So, no matter which of the solutions that you wind up using, one thing you should keep in mind is that whatever file name or primary key used to actually store the files needs to be easy to determine. And if you try to rely on the source of the data to name the file for you, that can cause problems, especially if they're naming scheme conflicts with some other source or they're using some character set that you might not want to have on the system.

Polling Question #2



Polling Question #2

How much malware does your organization collect (or soon hope to collect) each day?

1) Less than 100 per day
2) Up to 1000 per day
3) Up to 10,000 per day
4) Up to 100,000 per day
5) Over 100,000 per day

Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar

18

**018 Presenter: Okay that's going to be our third polling question we're going to launch now folks. And the question we'd like to ask is, "How much malware does your organization collect or soon hope to collect each day?" So, we'll give you another fifteen seconds to vote for that. And while we're waiting for that, Brent, a question from Dominic came in asking, "Can you use multiple database solutions for storing data?"

Presenter: Definitely. It might make sense to have a relational database for the surface level analysis or some of the static analysis that we'll get to in a little bit but store the indicators that you get from a runtime analysis system, since those tend to be so much larger, you might want to put those in a Hadoop or some other unstructured data system. So, that's definitely an idea.

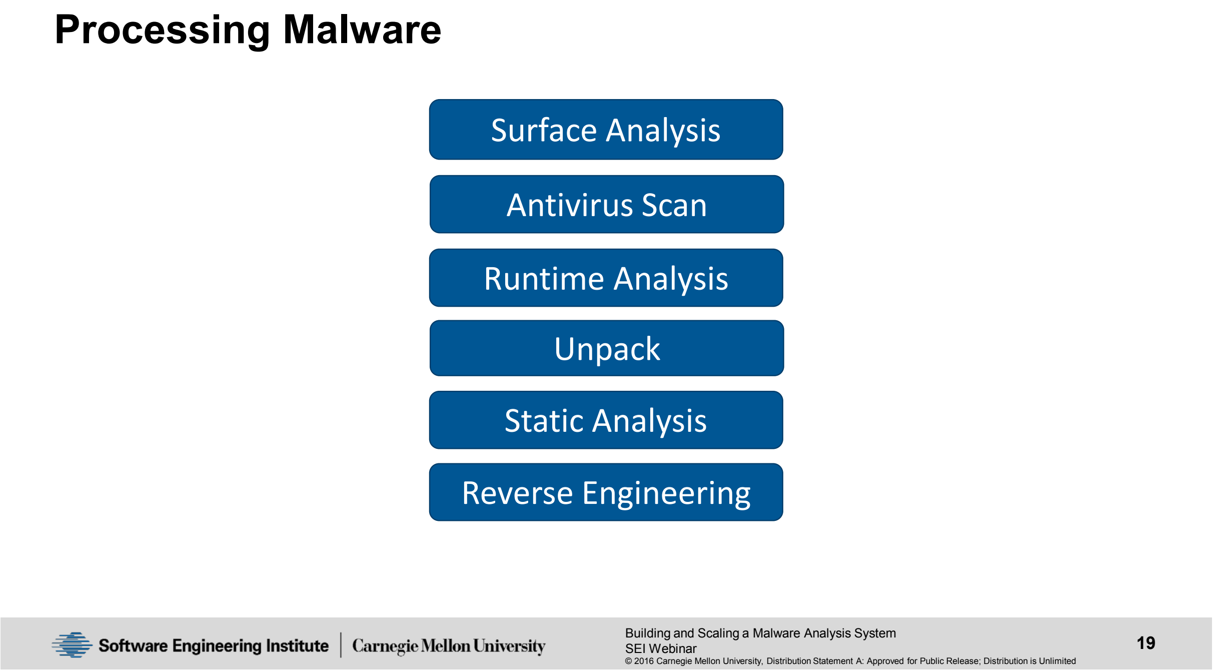
Presenter: Great. And before we go back to you, I'll just get a quick results from the poll. We had a fifty percent less than a hundred per day, twenty-nine percent up to a thousand, four percent up to ten thousand, four percent up to a hundred thousand, and thirteen percent over a hundred thousand.

Presenter: All right, so as with a lot of things when we're talking about scale, it's all relative. And so, that's interesting. A hundred thousand, until you start working with something like this, you don't realize that there are eighty-six thousand, four hundred seconds in every day. And it's not really an issue normally. But when you're doing something and you see people, fourteen, thirteen percent who say that they're getting a hundred thousand things or more a day, that's more than one every second.

Presenter: Right.

Presenter: And that can-- when you're dealing with that, and if you're dong dynamic analysis or something where it's taking ten minutes to do that run, you're adding up a lot of compute cycles in there. And it just really adds up.

Processing Malware



Processing Malware

- Surface Analysis
- Antivirus Scan
- Runtime Analysis
- Unpack
- Static Analysis
- Reverse Engineering

Software Engineering Institute | Carnegie Mellon University

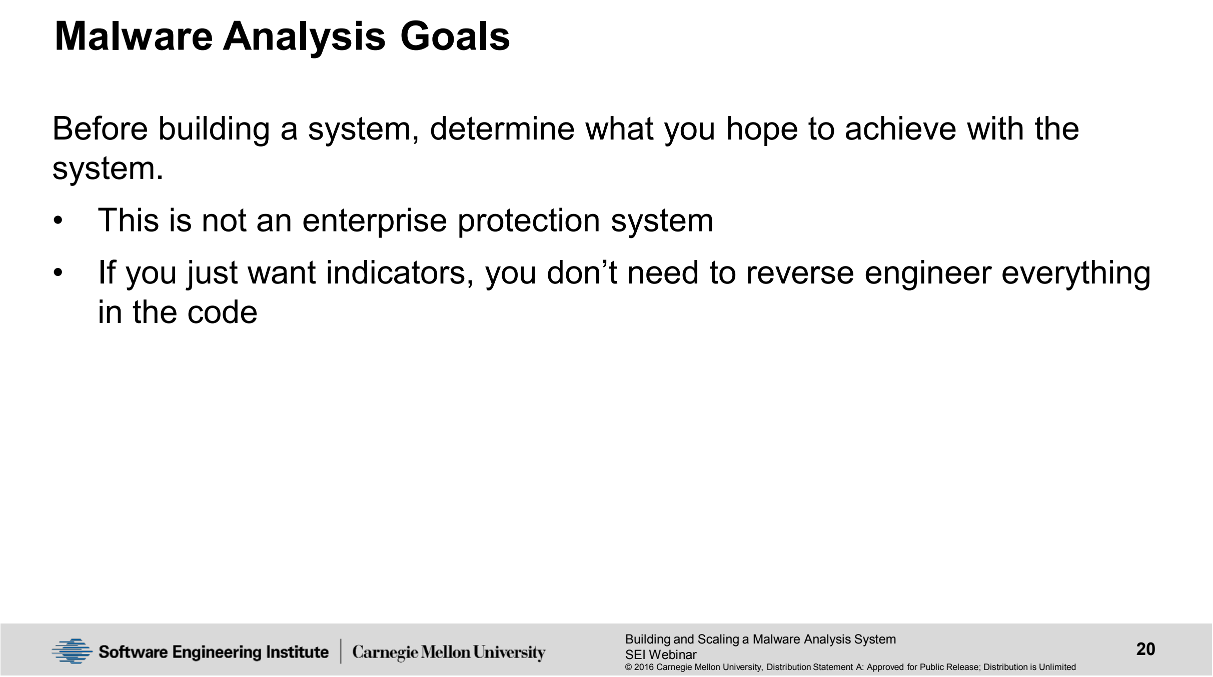
Building and Scaling a Malware Analysis System
SEI Webinar
© 2016 Carnegie Mellon University, Distribution Statement A: Approved for Public Release; Distribution is Unlimited

19

**019 All right. So, with the processing malware, which now that we've acquired the data, we've stored the data, we actually want to do something with it. The tasks I have here are not indicative of any particular taxonomy. They're convenient placeholders here so that we can break things up into some logical separation and discuss them in pieces. Some people might consider things like static analysis and reverse engineering to be the same thing. I'm not going to argue that point or anything. I'm just going to say this is some processes. And we're going to discuss some like this.

They're not in any particular order. It just seemed putting them in an easy hard order or a sequential order just didn't seem quite right here. But I will, towards the end of the presentation, have a slide that shows a possible ordering as far as a hierarchy for processing.

Malware Analysis Goals



Malware Analysis Goals

Before building a system, determine what you hope to achieve with the system.

- This is not an enterprise protection system
- If you just want indicators, you don’t need to reverse engineer everything in the code

Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar

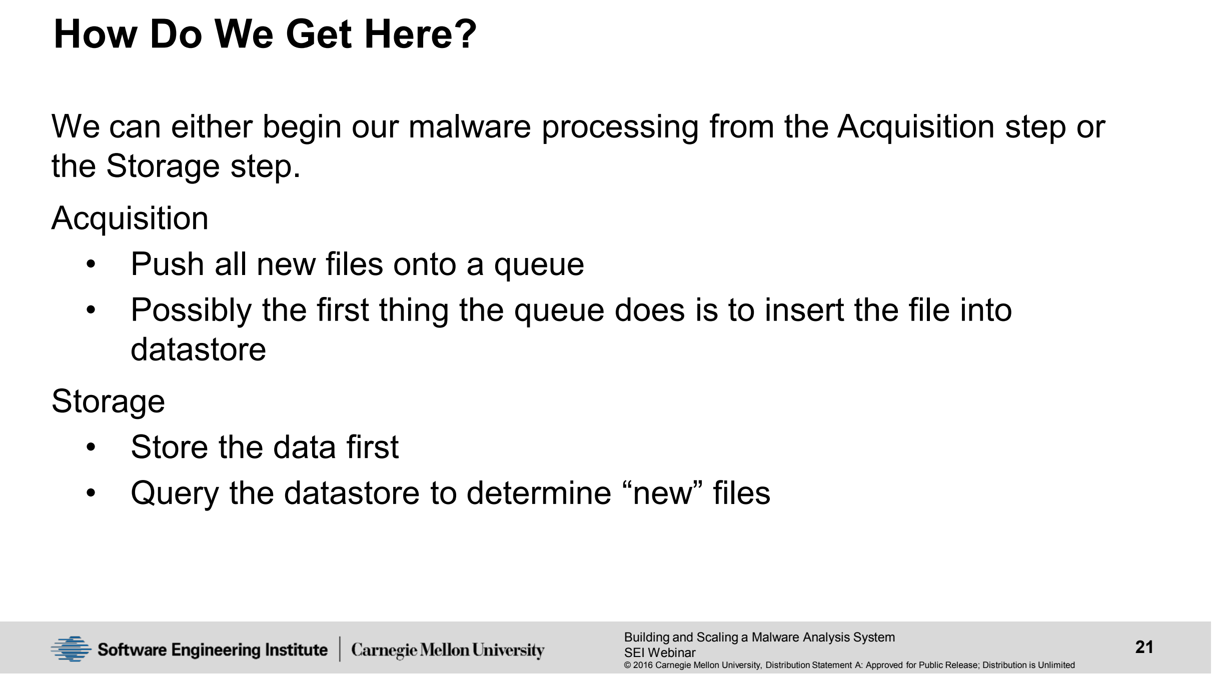
20

**020 So, before we actually do any, or talk about the processing, we need to figure out what it is the processing goals are because not all malware processing systems are created equal. Not everybody is trying to achieve the same goal. It might be necessary to perform all the tests I listed in that slide. You might want to-- but if all you really want to do is get a list of some of the indicators, then you might not need to reverse engineer everything. If you need a complete list of everything that that malware might do, then yes, you would need to go to reverse engineering.

So, the other point here is that the malware processing system is not like and enterprise protection system. It's not an AV system. You could-- in theory if you were to do some of the processing fast enough, and you didn't mind the latency involved or however long it would take to run things, you might be able to do that. But it's really not what this is all about. The enterprise protection systems, you are looking at potentially good and normal data with a few bad actors in there, some pieces of malware. With the malware analysis system, most of what you're expecting to see there is malicious. And you want to treat it as such. And take it apart and see what's going on.

So, an AV product or some other enterprise protection system product might come back and say, "This is Fred. I think Fred is bad." And that's all you'll get from them. But the malware processing system would come back and say, "Okay, this is Fred. Sometimes, he's called Frederick. Here's his height and weight. And here's some x-rays. And here's his DNA," and so on. And so, it's two very different problems that are being addressed here.

How Do We Get Here?



How Do We Get Here?

We can either begin our malware processing from the Acquisition step or the Storage step.

Acquisition

- Push all new files onto a queue
- Possibly the first thing the queue does is to insert the file into datastore

Storage

- Store the data first
- Query the datastore to determine “new” files

Software Engineering Institute | Carnegie Mellon University

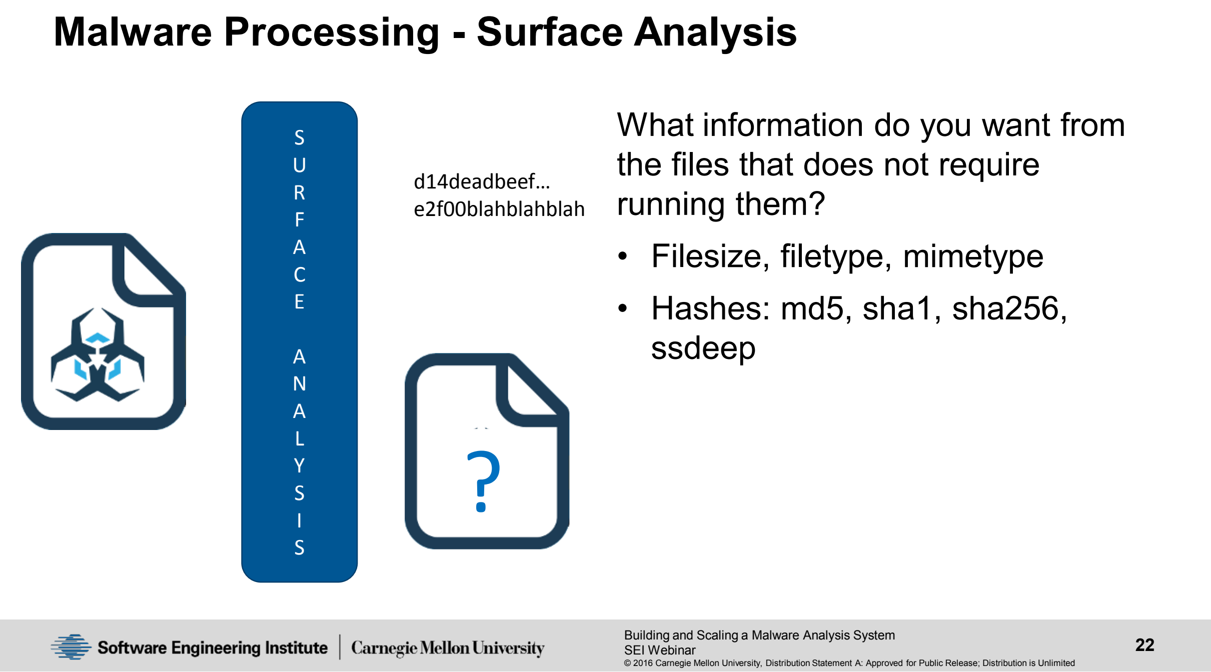
Building and Scaling a Malware Analysis System
SEI Webinar
© 2016 Carnegie Mellon University, Distribution Statement A: Approved for Public Release; Distribution is Unlimited

21

**021 And we need to point out how we get the data to the processing step. There's a couple different ways. One is you can have the acquisition system put all the files on to a queue and start working at them one at a time and then work from there.

Or you could have this storage system where you're storing the data first and then you have something else that periodically polls the storage system and says, "Okay, what's the new stuff that you've got since last time?" And the acquisition end of things, you might have a problem if you're getting lots of duplicates where you probably only want to process a file once. It's probably not going to change the results you get later on. And if you do that at the acquisition step, you're going to slow down your acquisition if you're going to a uniqueness check. But if you do it at the storage system, you'll have that uniqueness check. But you're going to have some additional latency because you'll have to wait until it does that polling step and see what's new.

Malware Processing - Surface Analysis



Malware Processing - Surface Analysis

What information do you want from the files that does not require running them?

- Filesize, filetype, mimetype
- Hashes: md5, sha1, sha256, ssdeep

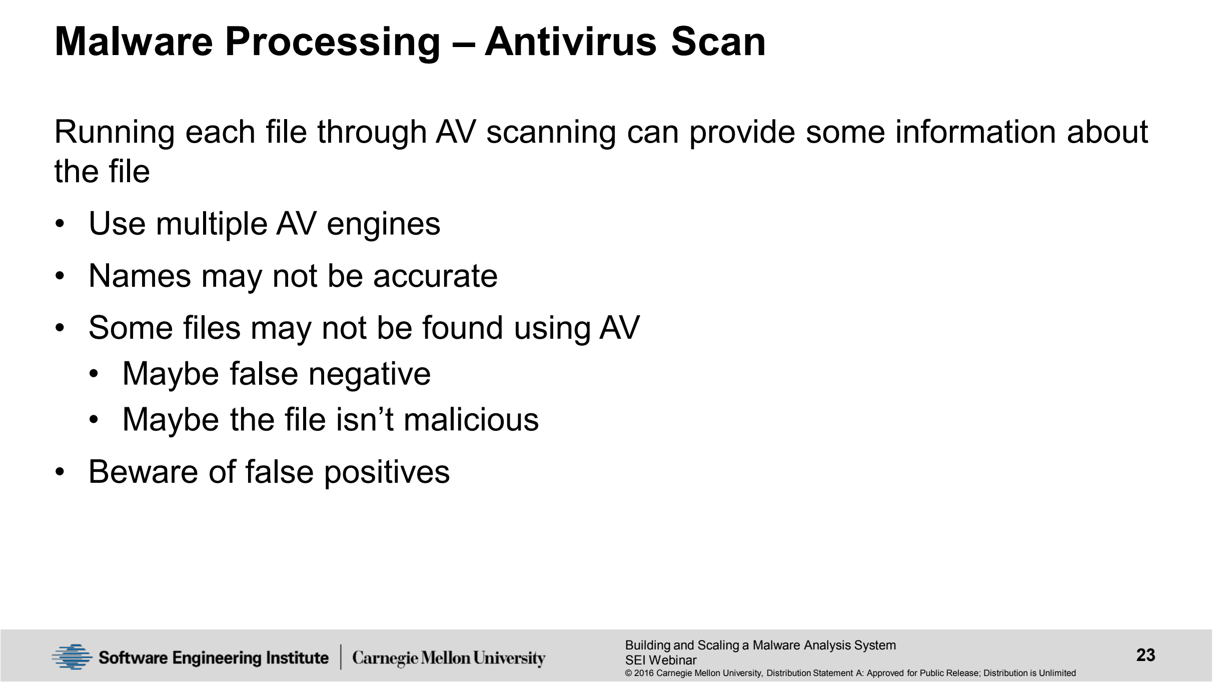
Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar
© 2016 Carnegie Mellon University, Distribution Statement A: Approved for Public Release; Distribution is Unlimited

22

**022 So, for surface analysis, surface analysis is the information about the file that's relatively easy to determine. It doesn't take a lot of resources, or time, or CPU. And it really usually doesn't tell you anything about what it does. It's usually focusing on what it is. So, md5, sha1, sha256, ssdeep, those are all hashing algorithms that will--the first three are this is exactly what this is based on these bytes. The ssdeep is a fuzzy hashing system. So, you can say it's close to some other file. The filesize, and filetype, and mimetype are usually heuristic-- well, the filetype and mimetype are heuristic based on what it looks like. And the filesize is obviously always going to be the same no matter what. But like I said, that will tell you what something is.

Malware Processing – Antivirus Scan



Malware Processing – Antivirus Scan

Running each file through AV scanning can provide some information about the file

- Use multiple AV engines
- Names may not be accurate
- Some files may not be found using AV
  - Maybe false negative
  - Maybe the file isn't malicious
- Beware of false positives

Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar

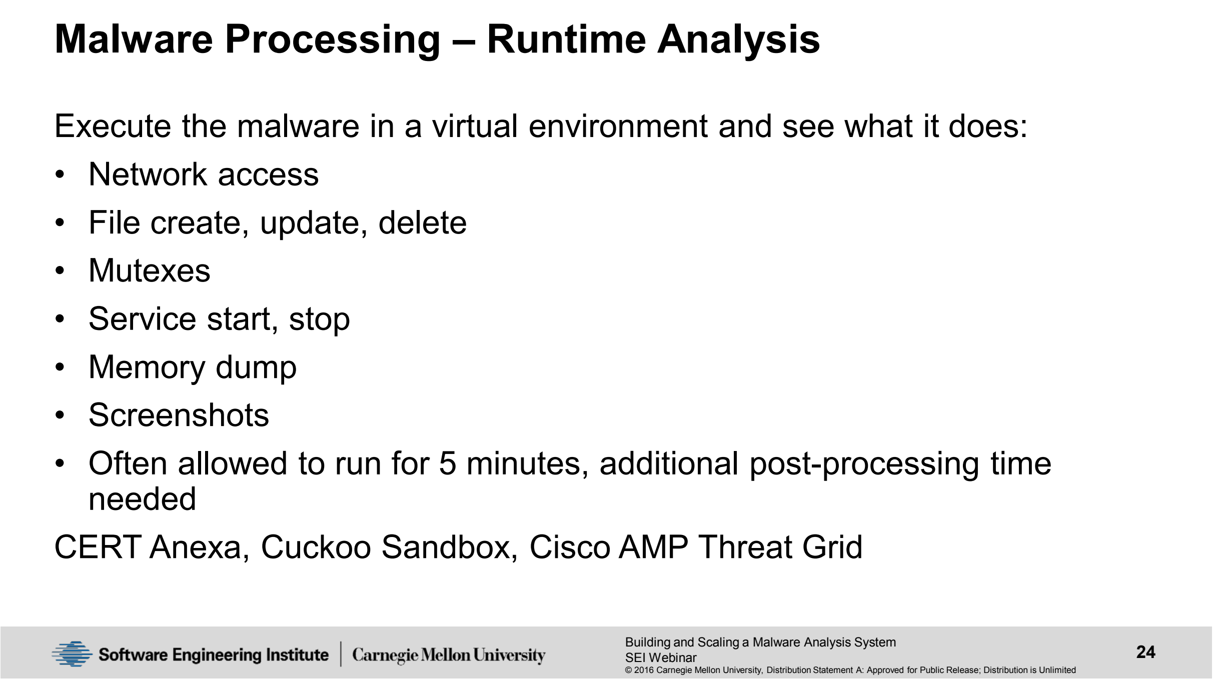
23

**023 The antivirus scanning will tell you what each individual antivirus product thinks it is. And it will come back, I think we're all familiar with antivirus products, where it will say okay this comes back as normal. Or this comes back as something malicious or maybe just suspicious. A lot of times it will say-- it won't know what it is or what to classify it as. But it will say that it is doing something unusual.

In a malware processing system, the name might very well be important. Most users, they don't care if it comes back, and it says, "This is Fred. This is Barney. This is Foobot." They just want it off their system if it's something that's malicious. Now, a processing system, if the AV system says it is Fred, and you have something oh, I know how to deal with Fred, I want to do this processing on that, if the name isn't right, if the AV system came back and it said it was Fred, but it was really Barney, you're going to have something different in your results than what you were expecting.

And by the same token, AV doesn't pick up everything. It's possible that the malware sample is too new for it to be in that signature database and be able to find it. Or it might be that it's so old that has been cycled out. And it's also possible that the file might not be malicious at all. And it's possible that when it comes back and says, "Okay, this is Fred," there have been cases in the news where the AV product has come back and said a required system file was a malicious piece of software. And removing it made the system unusable. So, you need to be careful on relying on the AV scans as ground truth. But they can be very useful so you can help communicate with other researchers.

Malware Processing – Runtime Analysis



Malware Processing – Runtime Analysis

Execute the malware in a virtual environment and see what it does:

- Network access
- File create, update, delete
- Mutexes
- Service start, stop
- Memory dump
- Screenshots
- Often allowed to run for 5 minutes, additional post-processing time needed

CERT Anexa, Cuckoo Sandbox, Cisco AMP Threat Grid

Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar
© 2016 Carnegie Mellon University, Distribution Statement A: Approved for Public Release; Distribution is Unlimited

24

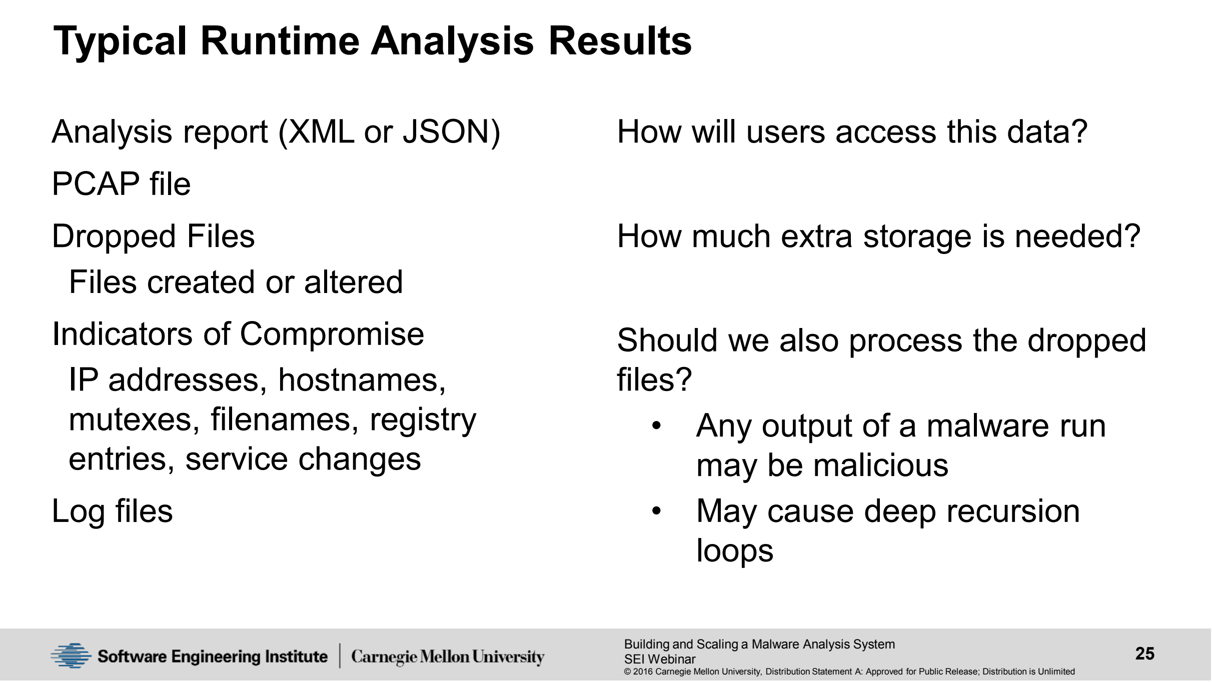
**024 So, after AV, we've got the runtime analysis where we're executing the malware in a virtual environment to see what it does. And in this case, it-- so, I've mentioned PE files before. But you can run anything through a VM as long as it has the executable on there that can process it. So, PDF files, you need something that can understand a PDF file. And it needs to be configured so that it uses the right thing because it's possible that maybe you have Adobe Acrobat Reader, but you actually want to test it against the Acrobat Writer product or something like that, just as an example there, if it's a PDF.

But all you're really doing in a runtime analysis system is essentially putting it on the system and double-clicking. And it will do whatever it's going to do. And you monitor what comes out of there, so any network accesses. You look and see what files are created, what files are updated or deleted. You can record what Mutexes where stored or were used in the processing, what services were started or stopped. You can dump the memory. You could take screen shots every second. And that can be important if all you're getting out of it is a dialogue box that says, "Press okay to continue." You could look back at the screenshots and know that okay, I have to do something about that.

And most of the runtime analysis systems that I'm aware of, by default, they'll run for about five minutes. Or you can configure them for whatever time you want them to. And they also have some post processing time after that and maybe some preprocessing time before that. So, my experience on average is you're going to be about ten minutes waiting for a system from the time you put the malware in until you get all of the results out of it.

And as we were talking about before, eighty-six thousand, four hundred seconds in every day. If you get over a hundred thousand files in a day, you're doing more than one a second. If you just had one a second, and you've got runtime analysis that takes ten minutes, which is six hundred seconds, that means you have to have six hundred VMs in there just to handle eight-six thousand, four hundred files at that rate. And the more the multiples of that, you have to have more. So, you have to keep this in mind when you are getting that set up, just how many of these systems you're going to need to keep up with the data that you've got.

Typical Runtime Analysis Results



Typical Runtime Analysis Results

Analysis report (XML or JSON)

PCAP file

Dropped Files

Files created or altered

Indicators of Compromise

IP addresses, hostnames, mutexes, filenames, registry entries, service changes

Log files

How will users access this data?

How much extra storage is needed?

Should we also process the dropped files?

- Any output of a malware run may be malicious
- May cause deep recursion loops

Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar
© 2016 Carnegie Mellon University, Distribution Statement A: Approved for Public Release; Distribution is Unlimited

25

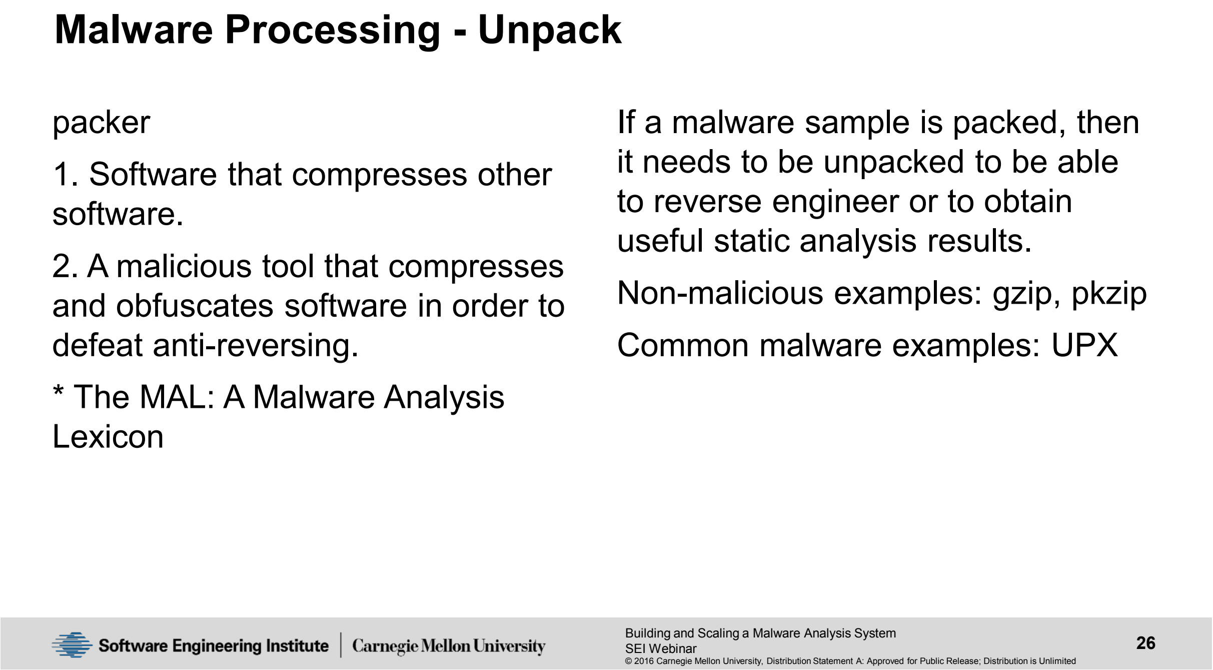
**025 The analysis results, we've got-- usually, you're going to have some sort of report in XML or JSON. You'll probably have a PCAP file that will tell all the networking stuff that's going on. The drop files, I mentioned things that are created or updated, the indicators of compromise, any IP addresses, hostnames, mutexes, filenames, all of these things that are changing that you can put in your database and say this is-- this file did this. And that might mean that this file is malicious.

Any of the log files that are there, and when you're storing them, you need to consider how the users are going to access this data. I've mentioned that with the possibility of having multiple databases or having the indicators being in one database. It's going to be however they want to use that and access it to meet their needs and try to compare different malware samples to see where you're going with it.

You need to keep in mind that you're going to need extra storage. All this stuff takes space, drop files in particular. If you have something that is a piece of malware that is going to infect a bunch of system files, you need to be prepared to handle potentially dozens or hundreds or possibly even thousands of files that have just had minor changes made that will inject that malware.

And then there's also the question of whether you should process the drop files because anything that has been created or altered by a piece of malware, it's very possible that it is itself malicious. And it might actually be something different than what had initially infected the system. The problem with that is, if you keep doing that, you're going to run out of space, especially if you have something like a polymorphic virus that everything it writes is going to have a different signature. And so, you might have like two hundred--every time you run one, you get two hundred new samples as far as your system is concerned.

Malware Processing - Unpack



Malware Processing - Unpack

packer

1. Software that compresses other software.

2. A malicious tool that compresses and obfuscates software in order to defeat anti-reversing.

* The MAL: A Malware Analysis Lexicon

If a malware sample is packed, then it needs to be unpacked to be able to reverse engineer or to obtain useful static analysis results.

Non-malicious examples: gzip, pkzip

Common malware examples: UPX

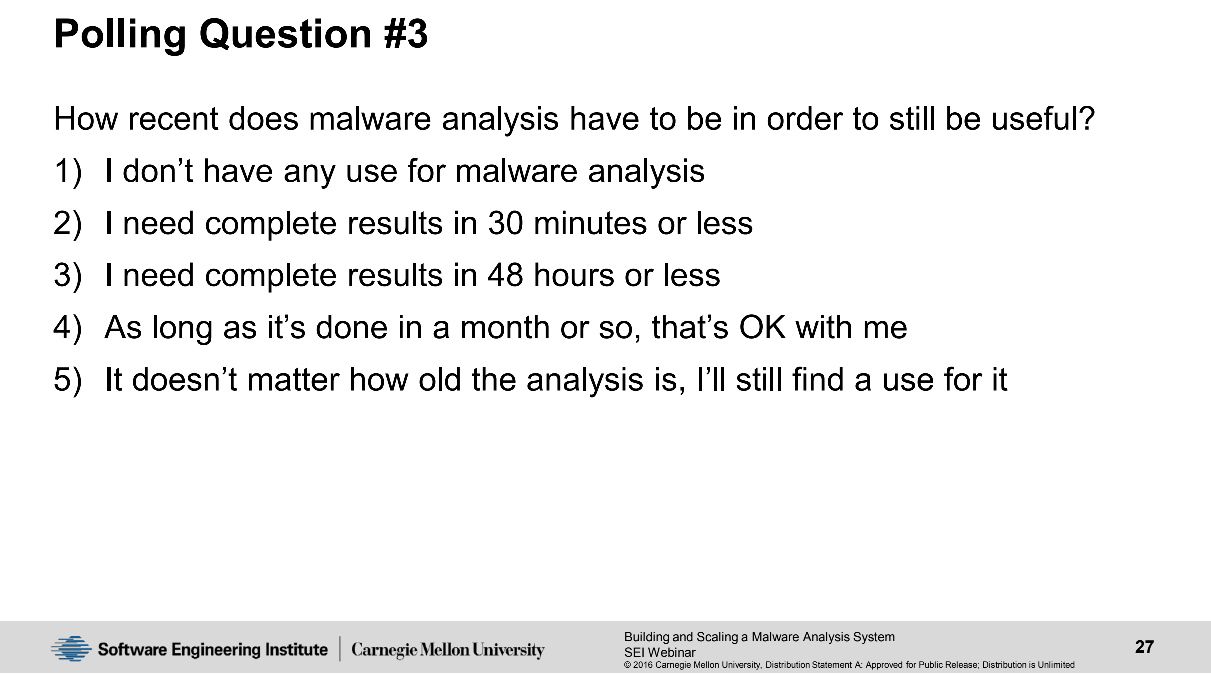
Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar

26

**026 The unpacking, so some malware is packed in order to obfuscate what it is that it does. The system needs to have some way of unpacking it when it's running so that it can run properly. But if everything is compressed, then it makes it that much harder to try and reverse engineer it to try and figure out how it is and go with that. So, there are some known packers. UPX is the most prevalent one in the case of malware that I'm aware of. And it's usually-- it's just a simple matter of using UPX-D, and you've got the original file. Although, it's possible that you may have had the file packed more than one time.

Polling Question #3



Polling Question #3

How recent does malware analysis have to be in order to still be useful?

1) I don’t have any use for malware analysis
2) I need complete results in 30 minutes or less
3) I need complete results in 48 hours or less
4) As long as it’s done in a month or so, that’s OK with me
5) It doesn’t matter how old the analysis is, I’ll still find a use for it

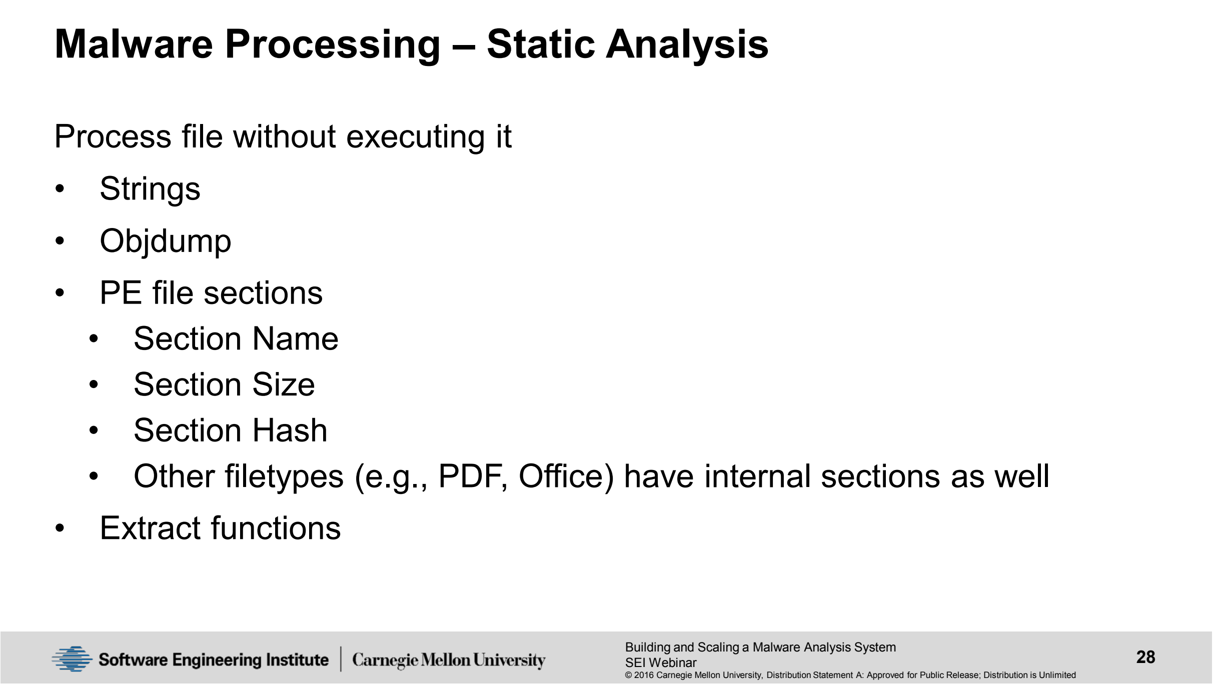
Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar
© 2016 Carnegie Mellon University, Distribution Statement A: Approved for Public Release; Distribution is Unlimited

27

**027 Presenter: So, we're going to go to our fourth and final polling question, which we'll launch here now. And what we'd like to know is how recent does malware analysis have to be in order to still be useful. Brent, I know you've got a couple more slides to go. We're down to about nine minutes. We're going to let you keep on going. And then we'll chime in with these results.

Malware Processing – Static Analysis



Malware Processing – Static Analysis

Process file without executing it

- Strings
- Objdump
- PE file sections
  - Section Name
  - Section Size
  - Section Hash
  - Other filetypes (e.g., PDF, Office) have internal sections as well
- Extract functions

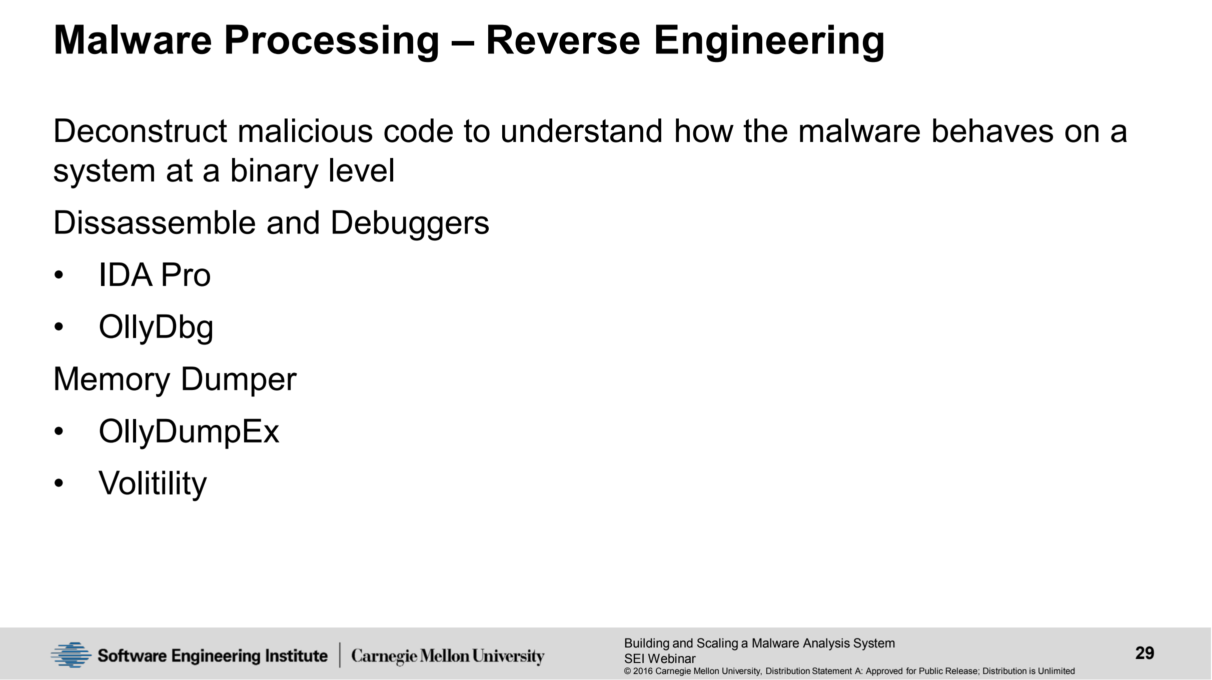
Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar

28

**028 Presenter: All right, so we'll talk about dynamic analysis, runtime analysis. We've got static analysis, which is where we're processing files without executing. And I mention these after unpacking because if you have a file that's packed, it doesn't make any sense do-- or it doesn't make a lot of sense to do static analysis on the packed file. You want it to be unpacked so that you get usable strings and usable objects so that if you've got a PE file, internally, it's organized by sections. And you can get names and sizes and maybe even hashes of each of those sections. And other file types like PDF or Office documents, those all have their own internal sections as well. And you might want to do that kind of static analysis for those. And if you have something that-- PE files that have functions, you could possibly extract those as well.

Malware Processing – Reverse Engineering



Malware Processing – Reverse Engineering

Deconstruct malicious code to understand how the malware behaves on a system at a binary level

Dissassemble and Debuggers

- IDA Pro
- OllyDbg

Memory Dumper

- OllyDumpEx
- Volitility

Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar

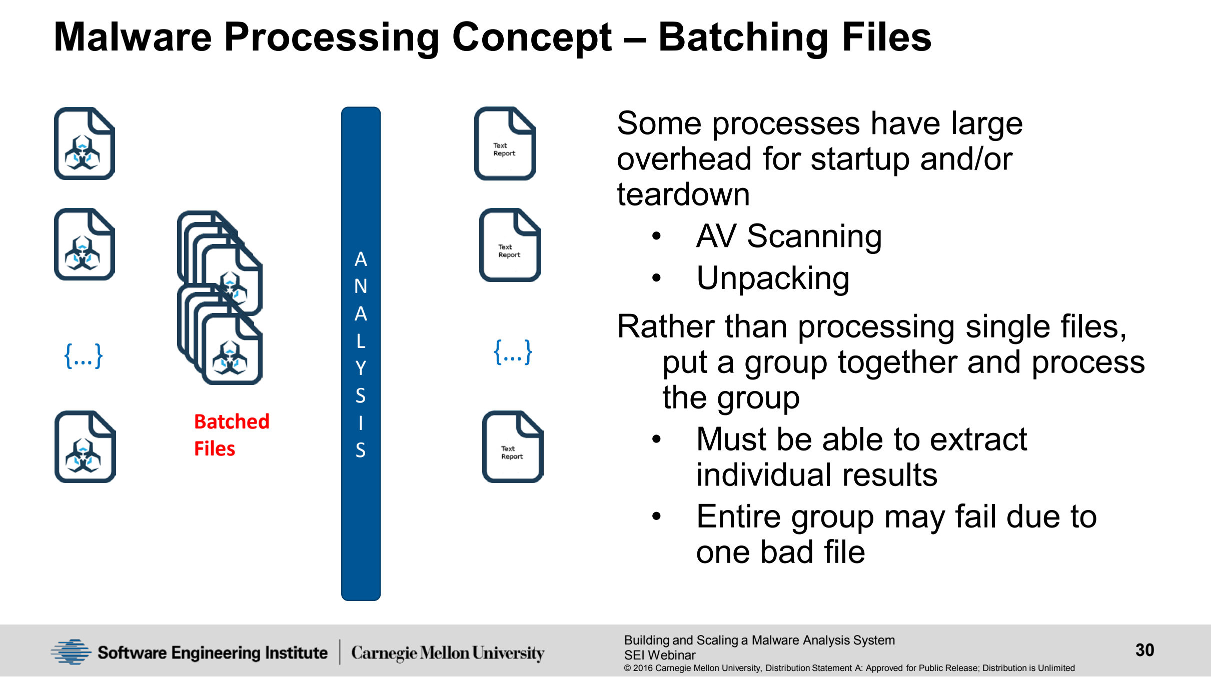
29

**029 I won't go into too much about reverse engineering simply because automating reverse engineering is very tricky. You can do a basic reverse engineering run with something like IDA Pro and get a first pass at it. But as I was saying with the packing, the malware authors are trying to obfuscate things as best as they can. They want to make it hard for somebody to come in and figure out what it is that software is doing. And there's tricks that can be used that will make it so that whatever you get out of IDA Pro or any other disassembler is not going to be usable for just right out of the gate. You're actually going to have to go in and manually go through and find out what that is.

And it's possible, after doing a number of these analyses and figuring out what malware family it is, that you can get a system to consistently automate for that particular family. But it's tricky to try and do that for everything all the time.

And so, I mentioned here memory dumpers. That's another thing where you might want to-- it's another tool that you can use to see what's going on in RAM with the malware after it's been loaded.

Malware Processing Concept – Batching Files



Malware Processing Concept – Batching Files

Some processes have large overhead for startup and/or teardown

- AV Scanning
- Unpacking

Rather than processing single files, put a group together and process the group

- Must be able to extract individual results
- Entire group may fail due to one bad file

Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar

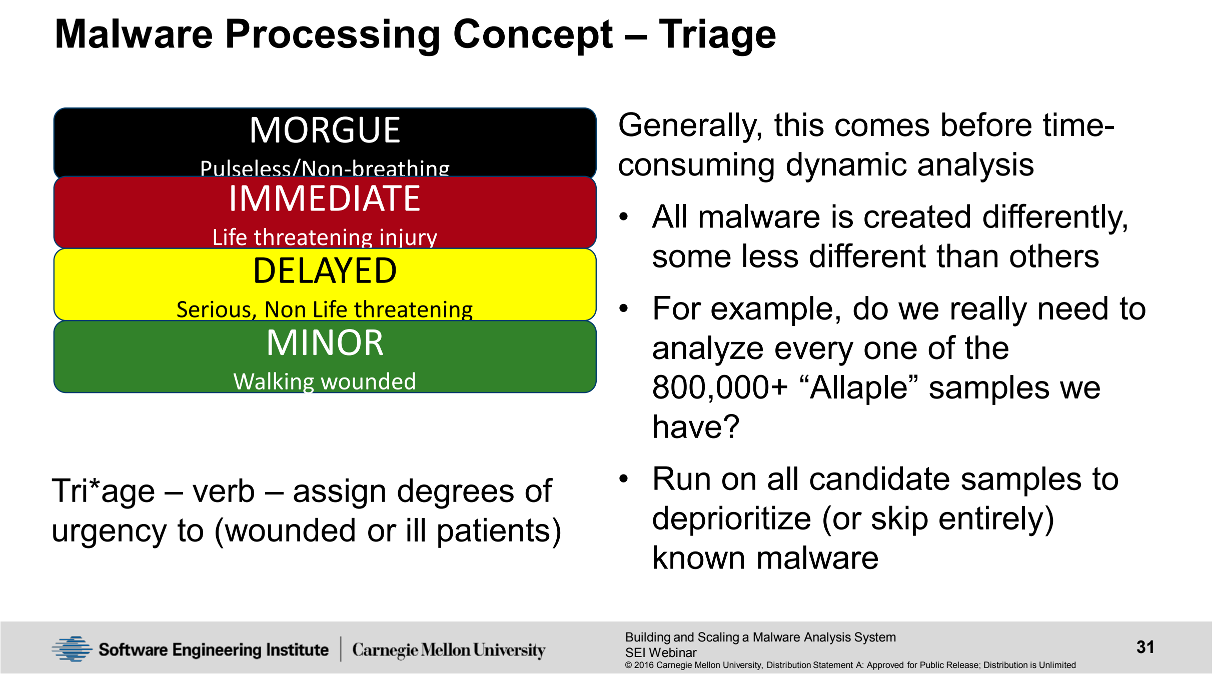
30

**030 So, one of the things with malware, there are a number of processes where the time to run the-- to set up a VM to run a sys-- the malware in it, it's too big compared to how long it actually takes to run it. So, it might take a fraction of a second to run it, but it might take several seconds to load it up.

And in a case like that, you might want to consider batching files together. I know people have tried that with AV scanning and with unpacking where they will bundle up a dozen or a hundred or some number of files and run them through. And then they have to parse the results back out so that this file had this AV signature and so on down the line.

Now, it's possible that whenever you're running those, again, they're malicious files. They're going to do odd things. So, if one of those files makes the system crash, then it's going to make that whole batch crash. And so, you have to have some recovery for that. But it is a way of improving the overall throughput of the system just by putting things together.

Malware Processing Concept – Triage



Malware Processing Concept – Triage

MORGUE
Pulseless/Non-breathing

IMMEDIATE
Life threatening injury

DELAYED
Serious, Non Life threatening

MINOR
Walking wounded

Tri*age – verb – assign degrees of urgency to (wounded or ill patients)

Generally, this comes before time-consuming dynamic analysis

- All malware is created differently, some less different than others
- For example, do we really need to analyze every one of the 800,000+ “Allaple” samples we have?
- Run on all candidate samples to deprioritize (or skip entirely) known malware

Software Engineering Institute | Carnegie Mellon University

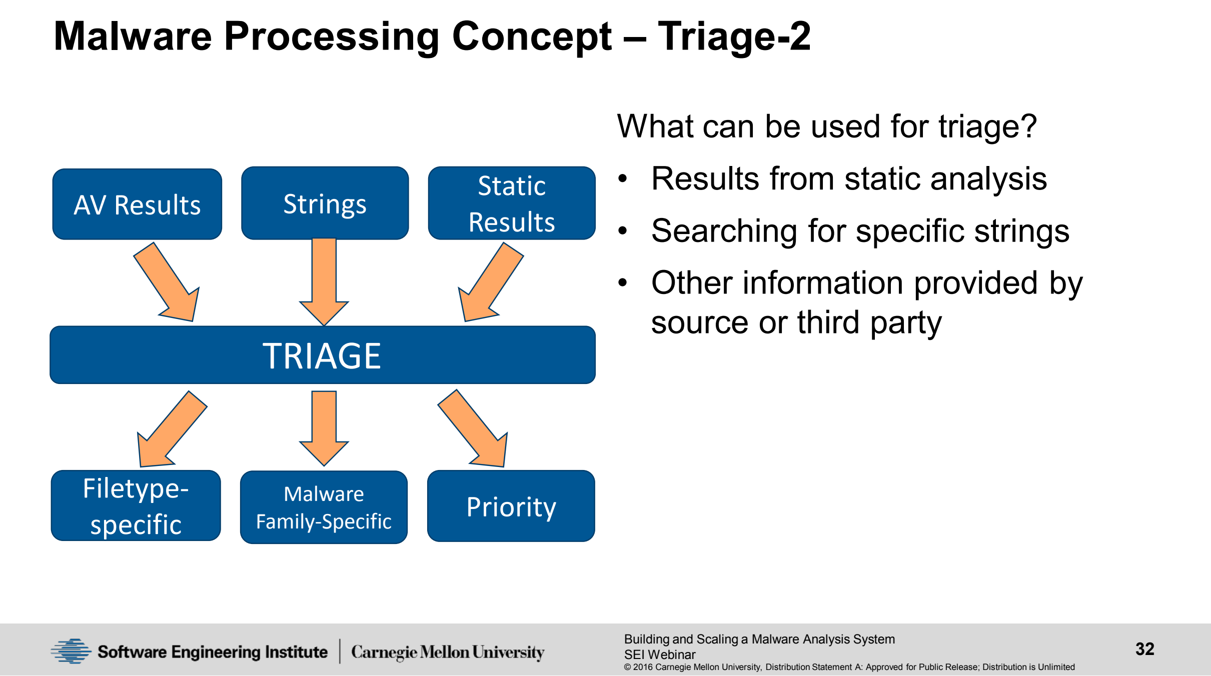
Building and Scaling a Malware Analysis System
SEI Webinar
© 2016 Carnegie Mellon University, Distribution Statement A: Approved for Public Release; Distribution is Unlimited

31

**031 Another thing you can do, if you are not necessarily concerned with getting everything about every file that you have come in, you could do kind of a triage or a filter on the incoming malware. So, I mentioned polymorphic malware before. We have one family from Allaple where we had over eight hundred thousand of them. And when we get eight hundred thousand and one, do we really need to do the exact same processing? And we're getting pretty much exactly the same runtime results and everything else. Do we really want to do that?

And if we have some way to restrict what goes through to the runtime analysis system based on some low hanging fruit, some easy to determine metrics like the AV family, or it matches this YARA signature, then we can either deprioritize or maybe skip entirely that piece of malware.

Malware Processing Concept – Triage-2



Malware Processing Concept – Triage-2

AV Results

Strings

Static Results

TRIAGE

Filetype-specific

Malware Family-Specific

Priority

What can be used for triage?

- Results from static analysis
- Searching for specific strings
- Other information provided by source or third party

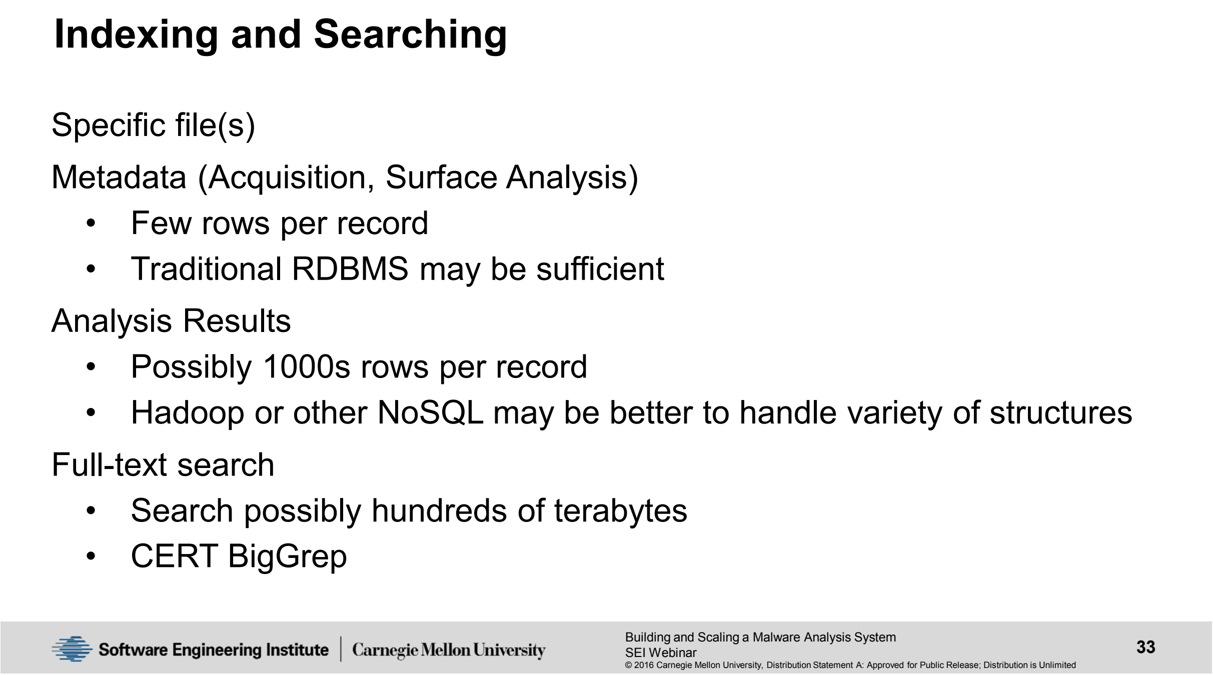
Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar
© 2016 Carnegie Mellon University, Distribution Statement A: Approved for Public Release; Distribution is Unlimited

32

**032 And we could use AV results, strings, results from static analysis. We could have those go into the triage system. And the output of there would be some notation or something where okay, we need to do some file type specific action on that piece of malware, or some family specific action that piece of malware. Or we can say it's a priority, lower priority. Or maybe it's a higher priority because of what kind of file it is. And we really want to get more information about that kind of family. And we can put that through the system and get that.

Indexing and Searching



Indexing and Searching

Specific file(s)

Metadata (Acquisition, Surface Analysis)

- Few rows per record
- Traditional RDBMS may be sufficient

Analysis Results

- Possibly 1000s rows per record
- Hadoop or other NoSQL may be better to handle variety of structures

Full-text search

- Search possibly hundreds of terabytes
- CERT BigGrep

Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar

33

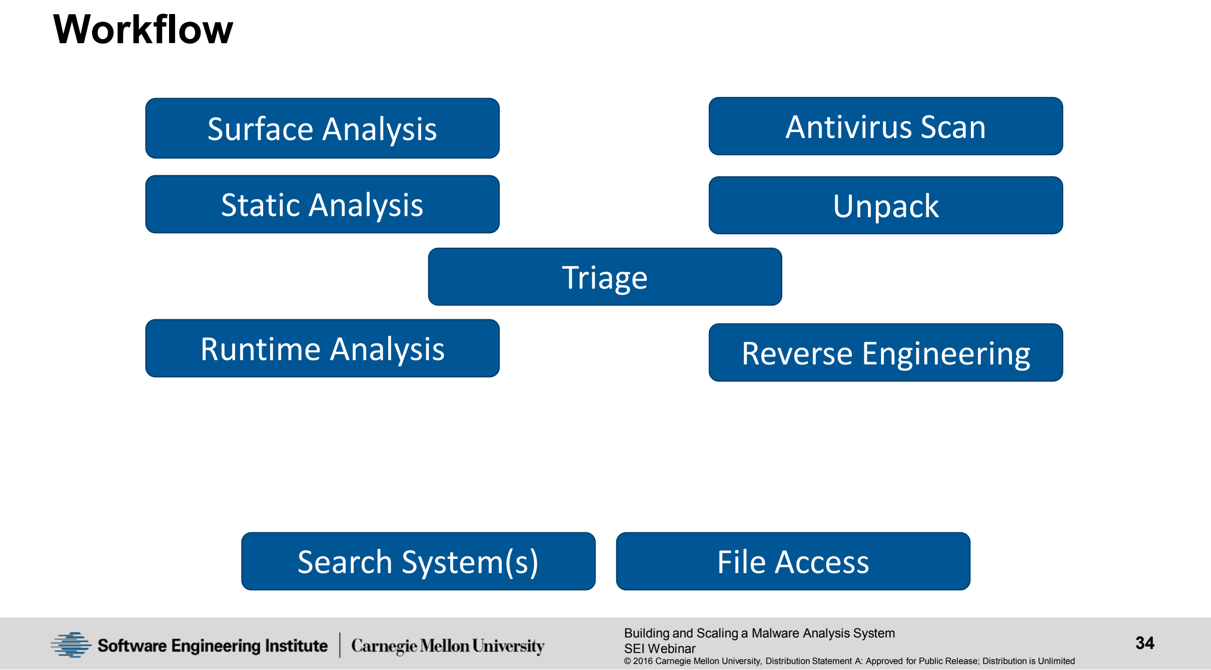
**033 Now once we've done all of the analysis, it's a matter of finding the data that we've got. So, there's a couple of different ways that people want to access data when they are trying to do their analysis. We've got the people who are looking for a specific file or set of files. In that case, they're usually coming up with a list of MD5s or sha256s and say, "Give me these." And that's pretty straightforward.

Then we have the metadata like the acquisition or the surface analysis where there's only one or two rows probably per record. And a traditional database system can be sufficient for that sort of thing.

The analysis results themselves, you might have thousands of rows of indicators if you've got something that is doing a network scan, for example, or that is infecting thousands of files. You're going to have a different row for each one of those things that comes through. And for that, when we have hundreds of millions of files, and we're talking thousands of rows for some of these records, that might just cause your database to roll over or just not go fast enough. And so, we might want to try a different solution for that.

And then we've got full text searching. Sometimes, people want to find I've got this long string that might be an array of data for encrypting or for doing a hash or something that is unique to a particular family of malware. And we want to search for that. But searching through using like the UNIX command grep is what we used to do way back in the day. And that would take days or eventually weeks to go through terabytes or now hundreds of terabytes of data. And we created a system called CERT BigGrep that will index those files and make them-- it won't be seconds like Google, but it would be certainly in minutes to be able to get candidates to say this is the files that you want to look for if you're looking for this particular string.

Workflow



Workflow

Surface Analysis

Antivirus Scan

Static Analysis

Unpack

Triage

Runtime Analysis

Reverse Engineering

Search System(s)

File Access

Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar

34

**034 And here's the workflow that I mentioned before. I don't have any arrows. But pretty much, we're looking at if you have your surface analysis and AV scanning and static analysis, and if you've got packed malware, those can all feed into a triage or filtering system. And then from there, we can just determine what kind of runtime analysis or reverse engineering we want to do. And all of those results can get put into the system for searching and for file access.

Questions?

Questions?

Software Engineering Institute | Carnegie Mellon University

Building and Scaling a Malware Analysis System
SEI Webinar
© 2016 Carnegie Mellon University, Distribution Statement A: Approved for Public Release; Distribution is Unlimited

35

**035 Presenter: So, Brent, thanks for the presentation today. Folks, we know it's at two thirty. But we wanted to work in just one question if we could wrap up, if it's answerable by you Brent. From Raphael asking, "Is there a separate team doing vulnerability research at CMU providing your team with the malware they discover? Or is this something you guys are discovering on your own?"

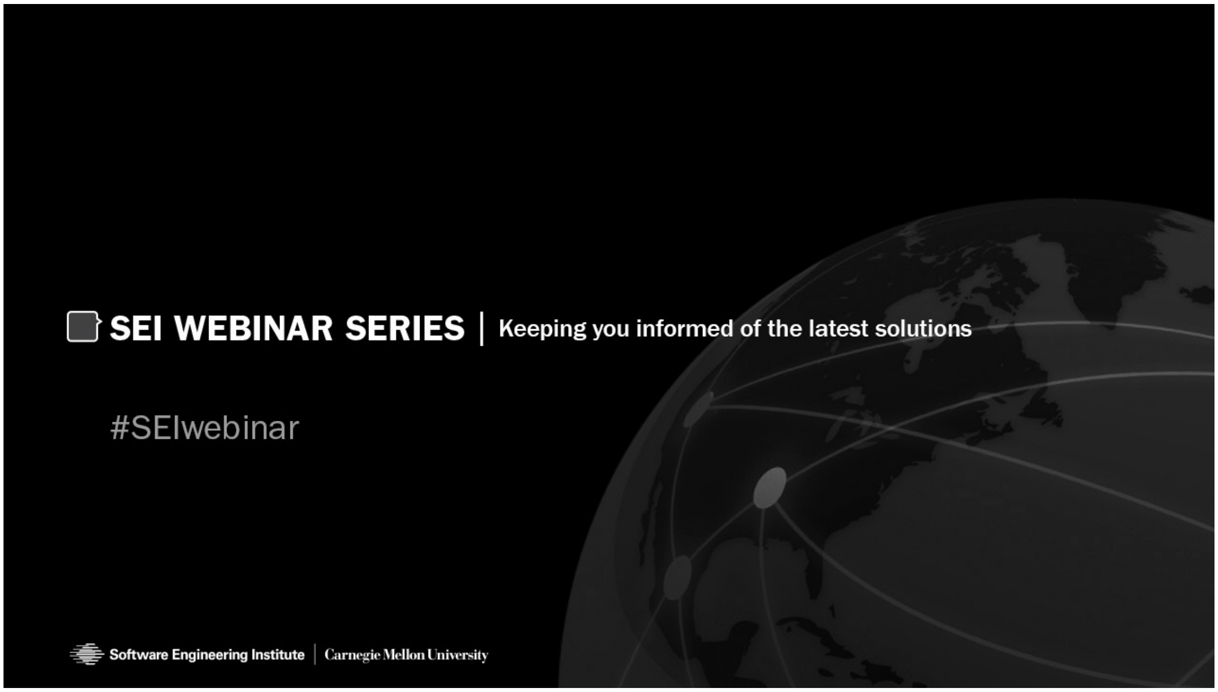
Presenter: So, we do have a vulnerability team at CERT that is doing vul discovery. But the vul discovery folks are not necessarily doing the same thing as the malware analysis folks.

Presenter: Right, they're getting in vulnerabilities.

Presenter: Right, they're looking at what breaks a system as opposed to what might be coming in on the system and being-- doing something malicious there. And as far as whether we're getting things from them, I'm not actually sure. We do have a number of sources. And I don't know what-- who all of them are.

Presenter: No, that's fine. Thanks again for your presentation. Folks, we know we're out of time today. We thank you very much for spending the hour with us. As a reminder, our next webinar will September 28th. We'll have a presentation on human factors from Jennifer Cowley and our science of cybersecurity group. So, you'll get an invite to that one. Thanks again everyone for attending. Have a great day.

SEI WEBINAR SERIES | Keeping you informed of the latest solutions



SEI WEBINAR SERIES | Keeping you informed of the latest solutions

#SEIwebinar

Software Engineering Institute | Carnegie Mellon University