Coordinated Vulnerability Disclosure Part 5

Table of Contents

SEI WEBINAR SERIES | Keeping you informed of the latest solutions 3

Carnegie Mellon University 3

Copyright 2016 Carnegie Mellon University 4

Coordinated Vulnerability Disclosure 4

Vulnerability Response 101 6

What is Vulnerability Response? 7

From discovery to the field 8

From discovery to the field 9

From discovery to the field 10

Why is vulnerability response important? 11

What is a vulnerability? 13

Exploits 14

Vulnerability Disclosure 15

Other Terms 16

The CVE ecosystem 18

Polling Question 20

Building Vulnerability Response Processes 21

Vulnerability Response Process Overview 22

Vulnerability Response Program 24

Building a Program – Initial Steps 25

Building a Program – Initial Steps 27

Polling Question 30

Maturing the Program 32

Communication Plan 36

External communications 37

Resources 39

Q&A 40

SEI WEBINAR SERIES | Keeping you informed of the latest solutions 42

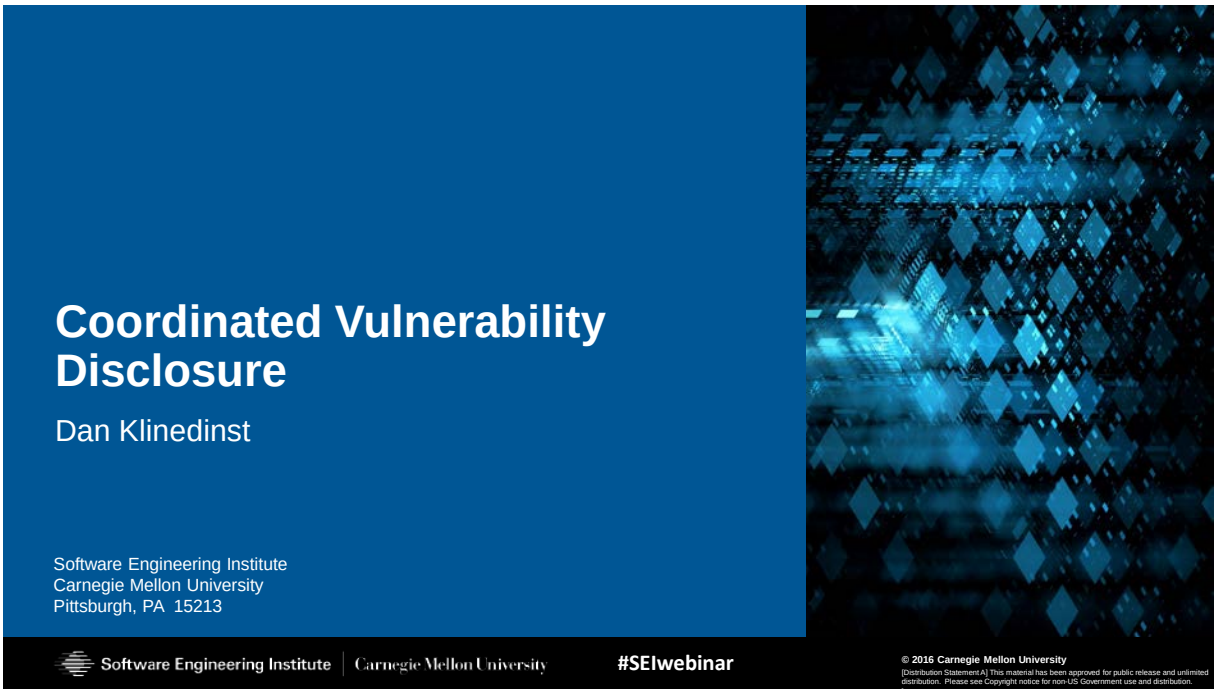
SEI WEBINAR SERIES | Keeping you informed of the latest solutions

Carnegie Mellon University

Carnegie Mellon University

Copyright 2016 Carnegie Mellon University

Coordinated Vulnerability Disclosure



**144 Presenter: All right, we're going to go on to our next talk which is

Coordinated Vulnerability Disclosure by Dan Klinedinst. And Dan is a vulnerability researcher at CMU CERT Coordination Center. He works performing vulnerability analysis of government and critical infrastructure assets. He currently is focused on researching security vulnerabilities in autonomous vehicles, edge computing platforms in embedded vehicles. Prior to this role, he was the technical lead for developing a national scale penetration testing program for a major U.S. government sponsor. Klinedinst is also the author of the Gibson3D Visualization tool and the technical architect for several international capture the flag events.

Just a reminder everybody, for those of you on the Twitter, make sure you're following @CERT_division and use the hashtag SEIWebinar. For anybody joining us new, there is a tab at the bottom of your console to download materials where you can find today's slide and other work from CERT and the SEI on our many topics today.

So, Dan, welcome, all yours.

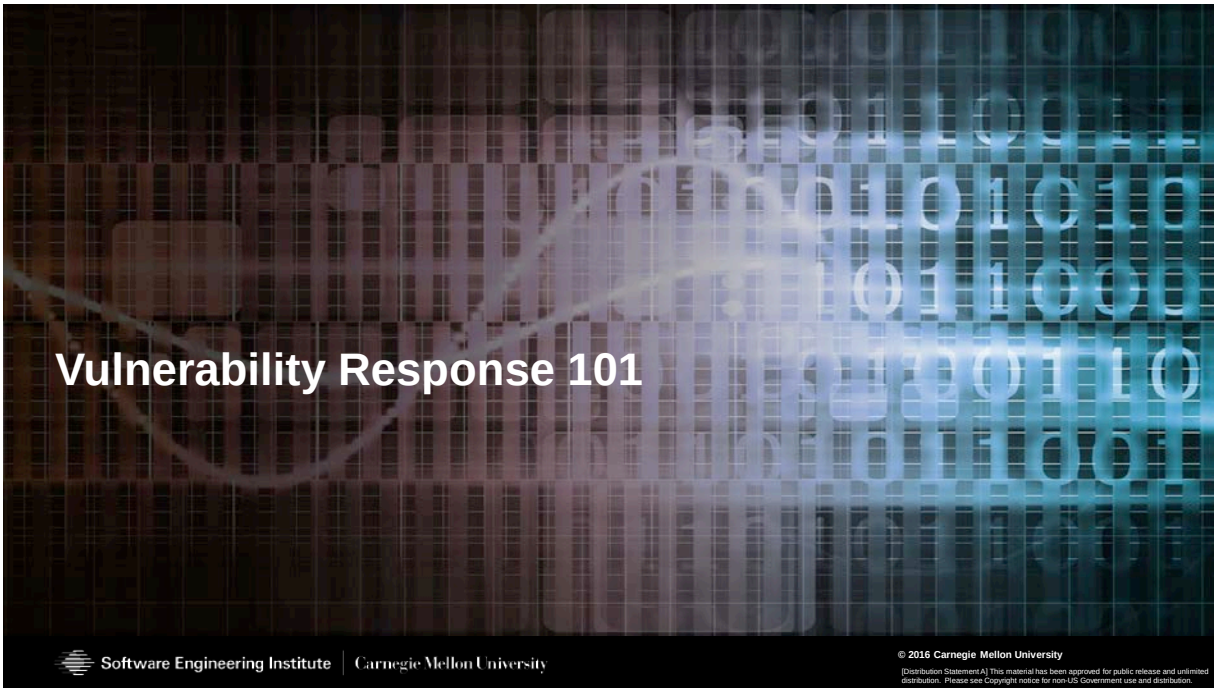
Presenter: Thank you, Shane. Hi, Mark.

Presenter: Hello there.

Presenter: So, as you might know, one of the main things that we do here at the CERT division is that we coordinate vulnerabilities that are found in software. And we have been doing this-- our very first vulnerability that we disclosed was in 1988. So,

we have a fair amount of institutional knowledge on the process.

Vulnerability Response 101



**145 And what we've discovered in the recent years is that, as has been mentioned several times today, and the Jeep hack kind of brings it out, increasingly software vulnerabilities and the coordination process are required for non-traditional software vendors. So, this is something that is fairly mature with the Microsofts, the Oracles, Google. It's still a very new thing when it comes to the Internet of Things, in automobiles, and other manufacturing fields.

So, we developed a course actually on building a product security incident response team, or PSIRT. And this is a very quick summary of what you would find out in that course.

What is Vulnerability Response?

Vulnerability response is the activity of responding to a vulnerability reported **in your organization's** product or service.

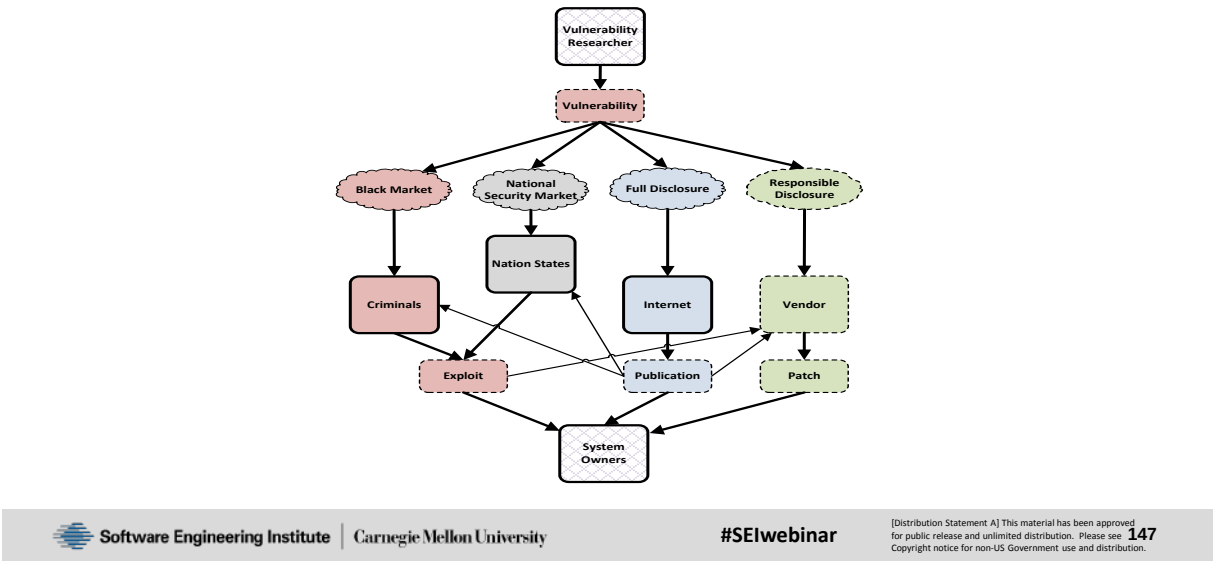
Vulnerability response is part of the larger ecosystem of product security; integrating security processes into the software development lifecycle.

This is not "vulnerability management" (or managing vulnerabilities of products that your organization has purchased or used)

**146 So first, what is vulnerability response? What we're talking about here is how your organization responds to vulnerabilities in products that you create. And this is part of the larger product-- larger ecosystem of product security that we've been talking about all day, sort of the tail end of that lifecycle in those cases in that we talked about security requirements. We talked about secure coding. We talked about DevOps. But inevitably, there will still be vulnerabilities that slip through all these processes. And it will be found by somebody outside of your organization after your product is released.

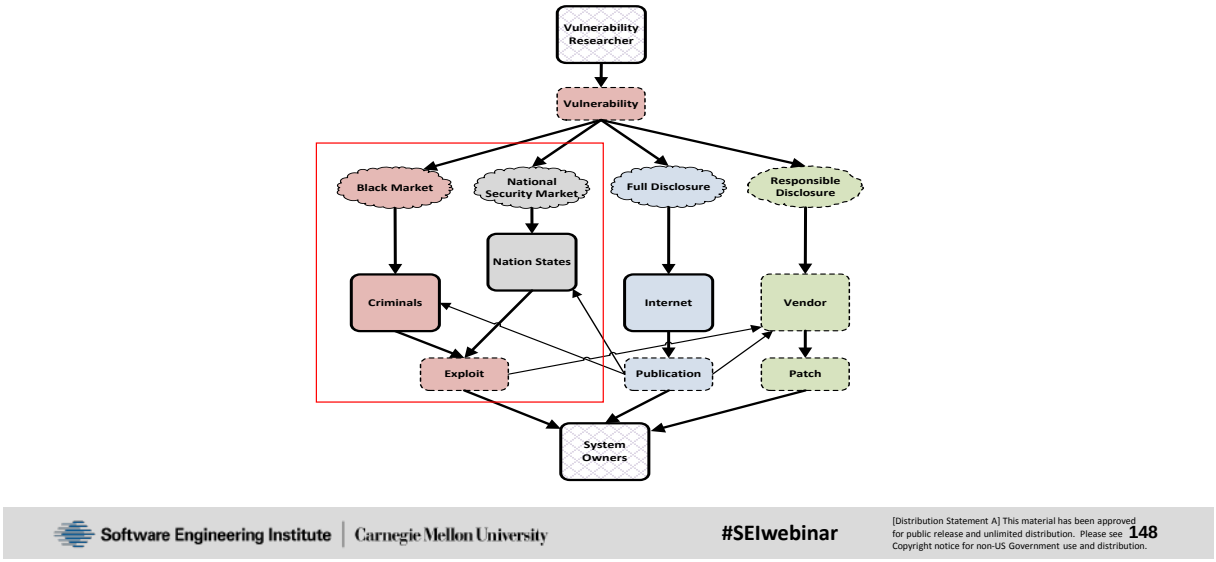
A quick note, we're not talking about vulnerability management, which is the management of vulnerabilities within your organization in software that you've purchased from other entities.

From discovery to the field



**147 So, there are a few different routes that a vulnerability can go after it's been discovered by a third party researcher. The two on the left of the slide, the vulnerability researcher can simply--

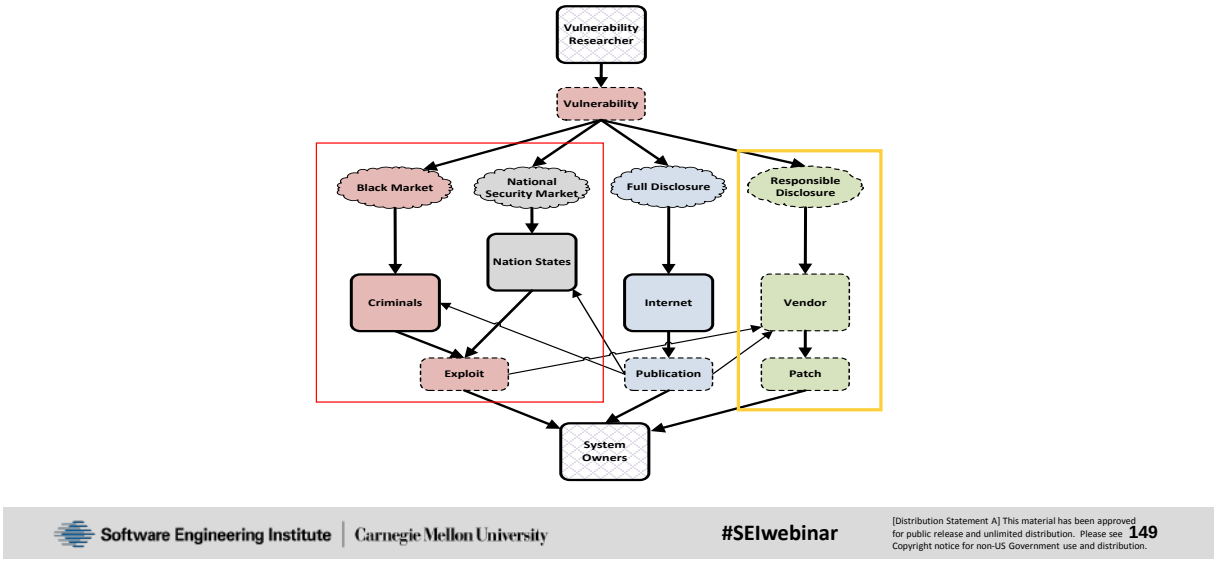
From discovery to the field



**148 Choose to sell the vulnerability to either the black market or the national security market, in which case, you, as the vendor, are never probably going to hear about the vulnerability unless it's discovered after it's already been used in the wild.

The other two options, full disclosure and responsible disclosure--

From discovery to the field



**149 What you want to do as a vendor is try to get people to the point where they're disclosing the vulnerabilities to you, the vendor, before the general public so that you have time to create a patch or remediate the vulnerability in whatever way you need.

Why is vulnerability response important?

Why is vulnerability response important?

- Being unable to be reached by a researcher (or CERT) will result in full disclosure of the vulnerability in your product.
- Slow or improper handling of a vulnerability may result in bad press, legal action, or harm to your customers.
- A good response process can
  - reduce the cost of fixes (by integrating it with your development cycle)
  - encourage researchers to test your software responsibly
  - produce good press about your company

**150 So, this leads directly into why is vulnerability response important. It's because you want the researchers to talk to you, the vendor, so that the problem can be addressed before the vulnerability becomes public. And one of the main reasons why people don't go through that process and disclose to the vendor is that they're unable to reach the vendor. So, often, the security researcher, or even us at CERT, have trouble reaching the manufacturer or getting an answer back. And that's one of the cases where a vulnerability ends up getting disclosed with no patch available or no remediation. Also, if your organization is slow or somehow does improper handling of vulnerability, it often leads to bad press, possibly legal action. And certainly harm to your customers should be the primary concern.

On the other hand, a good response, first of all, it reduces the cost of fixes because now that vulnerability response is embedded into your development lifecycle, you'll be prepared to handle them when they come up rather than having to suddenly drop everything that you were working on to address and urgent security fix. It also encourages researchers to test your software responsibly. So, you might think, "Wait, why do I want to encourage researchers to test my software for security vulnerabilities?" But actually, you should look at it as free security testing for your applications after they're already in the field. These are people, if they're reporting them to you, they're people who have not decided to sell them. they're not dumping them on the Internet with no warning. They want to work with you to responsibly handle these vulnerabilities. So, it helps you to work with them also.

And finally, of course, it can produce good press about your company. For example, we had Chris Valasek here earlier talking about his process with the Fiat Chrysler and the Jeep hack. Fiat Chrysler, although they had a security vulnerability, they were prepared with a patch before Chris's and Charlie's talks came out. They worked with Sprint soon after it was released to mitigate the worst effects of the attack. So, overall, this was a fairly new process to them. They handled it very well.

What is a vulnerability?

What is a vulnerability?

A vulnerability is:

"a set of conditions that allows an attacker to violate an explicit or implicit security policy. Vulnerabilities can be software defects, configuration or design decisions, unexpected interactions between systems, or environmental changes."

- from CERT definition

All software has vulnerabilities (and they increase with complexity)

Table 1 – Vulnerability density versus defect density measured for some software systems

Systems	Msloc	Known defects	Known defect density (per Ksloc)	Known vulnerabilities	V_{KD} (per Ksloc)	V_{KD}/D_{KD} ratio (%)	Release date
Windows 95	15	5000	0.3333	50	0.0033	1.00	Aug 1995
Windows 98	18	10000	0.5556	84	0.0047	0.84	Jun 1998
Windows XP	40	106500	2.6625	125	0.0031	0.12	Oct 2001
Windows NT 4.0	16	10000	0.625	180	0.0113	1.80	Jul 1996
Win 2000	35	63000	1.80	204	0.0058	0.32	Feb 2000

Alhazmi OH et al., Measuring, analyzing and predicting security vulnerabilities in software systems, Computers & Security (2006), doi:10.1016/j.cose.2006.10.002

**151 I'd like to talk about a couple of terms just to clarify what we're talking about here. A vulnerability is simply anything that can be done to a product that violates and explicit or an implicit security policy. So, sometimes of course, you have to make security tradeoffs in the design of your product. And that's fine as long as the customers are aware of the tradeoff and the risk that you're asking them to take. And to start with a lot of numbers down here at the bottom, all this really says is that the number of vulnerabilities tends to increase with the complexity of software. So, I've read that a modern vehicle has something like a hundred to a hundred and fifty million lines of code these days, about equal to a common operating system. So, that leads to potentially many vulnerabilities.

Presenter: Or as much, I've also seen a comparison with the amount of mouse DNA. Yes.

Presenter: Yes. The number of gene pairs in a mouse, I believe.

Exploits

Exploits

An exploit is:

"a piece of software or technique that takes advantage of a security vulnerability to violate an explicit or implicit security policy."

- from CERT definition

0-day vulnerability

- A vulnerability that is known by a 3rd party by the time it is publicly discovered (and no patch is available).

**152 So, an exploit, an exploit is a piece of software technique that takes advantage of the security vulnerability. And this leads directly to the definition of a zero-day vulnerability. Now, we recently blogged about the fact that it's very difficult to define zero-day vulnerabilities. So, just for conversational purposes, we're going to say that a vulnerability that gets disclosed and there is no patch available is a zero-day vulnerability regardless of whether or not there's an exploit for it yet.

Vulnerability Disclosure

Non-Disclosure

A vulnerability that may be reported to the vendor (or not), but is not disclosed to the public regardless if the vendor decides to fix the vulnerability.

Full Disclosure

A vulnerability that is disclosed to the public, so both the vendor and the public find out about it at the same time.

Coordinated Disclosure

Also called "responsible disclosure"

A vulnerability that is disclosed to the vendor, and after a reasonable period of time is disclosed to the public.

**153 And we've touched on this a little bit already, but a few common forms of vulnerability disclosure, first of all, there's non-disclosure. It sometimes happens that a vendor is informed about a security vulnerability. They go ahead and fix it. And it's never made public. This often happens when the company has hired third party security testers or if perhaps the person that found the vulnerability was covered by an NDA or something like that.

Full disclosure is simply the person that discovers the vulnerability dumps it on the Internet. And the vendor has no time to patch, no remediation ready. They find out about it the same time as everybody else.

And finally, our preferred route is coordinated disclosure, sometimes also called responsible disclosure. And that's a vulnerability that's disclosed to the vendor. And then after a reasonable period of time, it is also disclosed to the public.

Other Terms

Other Terms

Common Vulnerability Enumeration (CVE)

- A unique identification number for public vulnerabilities. Maintained by MITRE at cve.mitre.org
- Not a complete list of all vulnerabilities

Common Vulnerability Scoring System (CVSS)

- A method of determining the severity of a particular vulnerability. Maintained by FIRST at first.org/cvss

**154 A couple of other quick terms here you might hear, common vulnerability enumeration, or CVE. This is a unique identifier that is assigned to any public vulnerability. And it's not a complete list of vulnerabilities. There are certainly non-public ones. And CVE does not even attempt to cover all known public vulnerabilities.

Presenter: Does MITRE sort of restrict their view of what they're tracking in CVE?

Presenter: There's certainly a definition of what they consider a vulnerability which may-- not all vulnerabilities are going to be eligible for a CVE.

Presenter: Any restrictions like by industry or type of software or what they're focused on?

Presenter: No, they've done CVEs for hardware issues and more medical technologies and things like that. So, I think it has more to do with the type of vulnerability and the impact that it has and also just the number of them. So, we discovered Android apps. We discovered twenty-five thousand vulnerable Android apps with some automated testing. And that's just too many for the CVE process to handle. And then also the common vulnerability scoring system is a way to make a rough guess at the priority of a vulnerability based on the impact it has, how easy it is to exploit, etc.

The CVE ecosystem

Why people care about CVEs:

- Most security tools use the National Vulnerability Database as a feed to update their vulnerability scanners
- These scanners are widespread and used by security companies, internally at organizations, and by governments.
- When a vulnerability appears on a scanner, the client usually wants to fix it. If there is no fix, they will complain. For some clients (government, military) a vulnerability is a show stopper for implementation.

Image courtesy of Tenable.com

**155 And the reason these are important is because a lot of tools that are used to test the security in organizations, their sole number of vulnerabilities that they look for are ones with CVEs. So, if your vulnerability doesn't get a CVE, your customers may not be looking to see if they're vulnerable to it. And then they use a CVSS score to rank how important those vulnerabilities are to fix. So, in some regulated industries and government and the military, you actually can't have vulnerabilities on the network with CVSS scores above a certain level, or your system may be taken offline.

Presenter: Can you explain a little bit about how CVSS is calculated and how that might apply to cyber-physical systems?

Presenter: Sure, in fact, we had a recent blog post by myself about how CVSS applies to cyber-physical systems. And essentially, the base score for CVSS looks at what the impact is on integrity, confidentiality, and availability, and looks at what the access vector is, so if it's network or local, etc., how easy it is to exploit, whether authentication is required. And there's one or two other things that are measured, characteristics. And you can expand it beyond that. There's also what are called temporal and environmental scores, which attempt to let you customize CVSS to your own organization and your own risk.

It works for cyber-physical systems, but there are some things that aren't really covered by CVSS currently that might be applicable. So, in the example of the Jeep hack, we had the possibility of human safety being affected. Human safety isn't currently a characteristic that's scored in CVSS. So, that's one place where it may need to catch up a little bit with the emerging technologies.

Polling Question

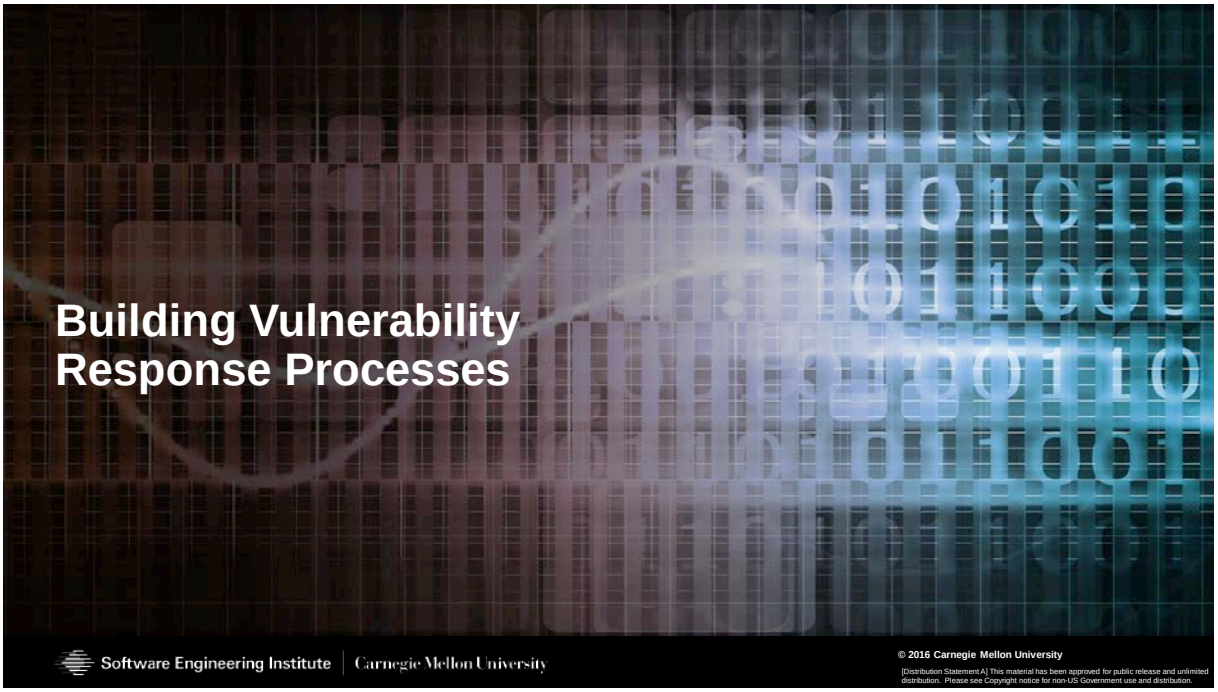
Does your organization have a vulnerability response program or Product Security Incident Response Center (PSIRT)?

A. Yes, fully operational
B. In the process of building one
C. These functions are handled ad hoc.
D. Not currently.

**156 Presenter: Okay, that's going to lead us to a quick polling question. And the question we'd like to know is, "Does your organization have a vulnerability response program or product security incident response center, or PSIRT?" The options are yes, fully operational, B, in the process, C, these functions are handled ad hoc, or D, not currently. So, we'll give the fifteen or twenty seconds to vote, Dan, and let you keep going. And I'll chime back in with the results here in a little bit.

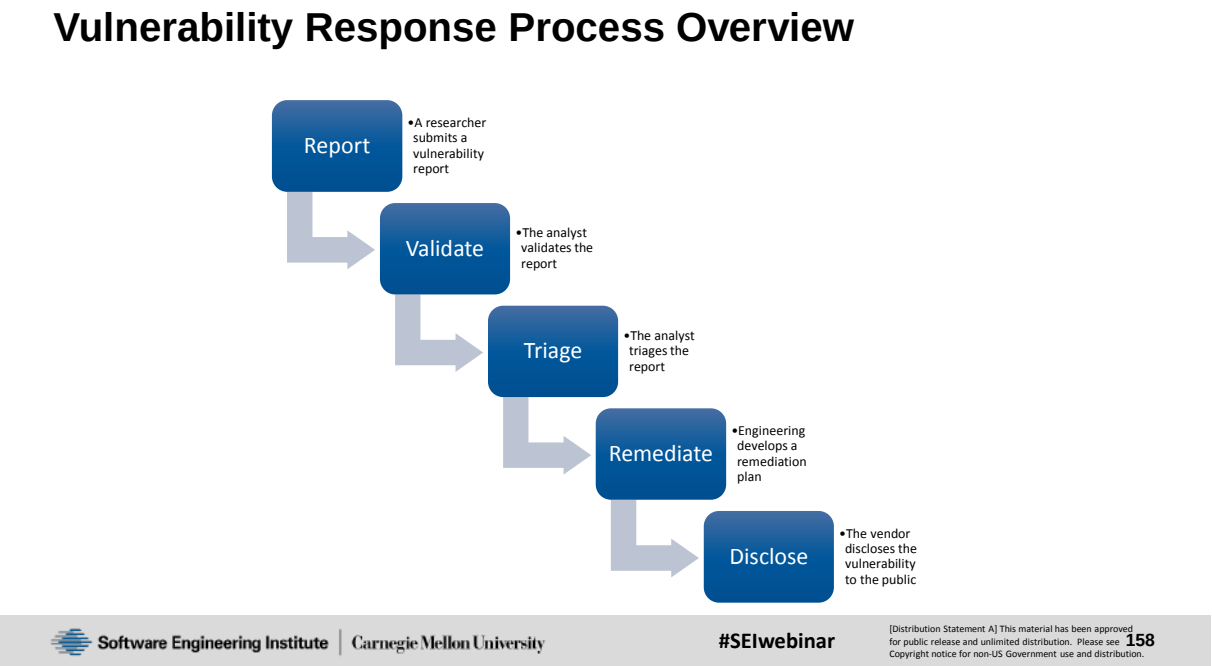
Presenter: All right, thanks.

Building Vulnerability Response Processes



**157 So, we're going to start talking about how to build a vulnerability response process if you don't already have one. And if you do already have one and have interesting information or questions, please make sure you type them in for us. We'd be happy to talk about them.

Vulnerability Response Process Overview



**158 So first, I'm going to run through an overview of the vulnerability response process as we see it. The first step is report. This is really an action that the security researcher is doing. They're reporting the vulnerability to your organization. But you, as an organization, need to be prepared to respond to that report. And we'll talk quite a bit about communication here in the next couple of slides.

The second step is validation. And you need to be able to verify whether or not a vulnerability that's reported is a real security vulnerability in your product. Certainly, you will get ones that either accidentally or intentionally are not valid security vulnerabilities. So, you need to be able to weed those out.

Then it goes into triage. If you decide you have a real security vulnerability, you need to look at the impact to your customers and how long it's going to take you to patch. Perhaps, there might be external factors that affect your timeframe, which you need to decide how to prioritize fixing that vulnerability. For example, if a researcher has already told you that they're going to present it at a certain conference, you'll know that that's your absolute deadline to have a patch out or you'll end up with essentially a zero-day, effectively.

The next step is to remediate. And this goes into your engineering team where they need to decide how to patch the security vulnerability. And you need to get it into your process or development. This goes into some of the things that we talked about earlier, the DevOps, your secure coding processes, to make sure that you don't introduce more security vulnerabilities by trying to fix the one that you found out about.

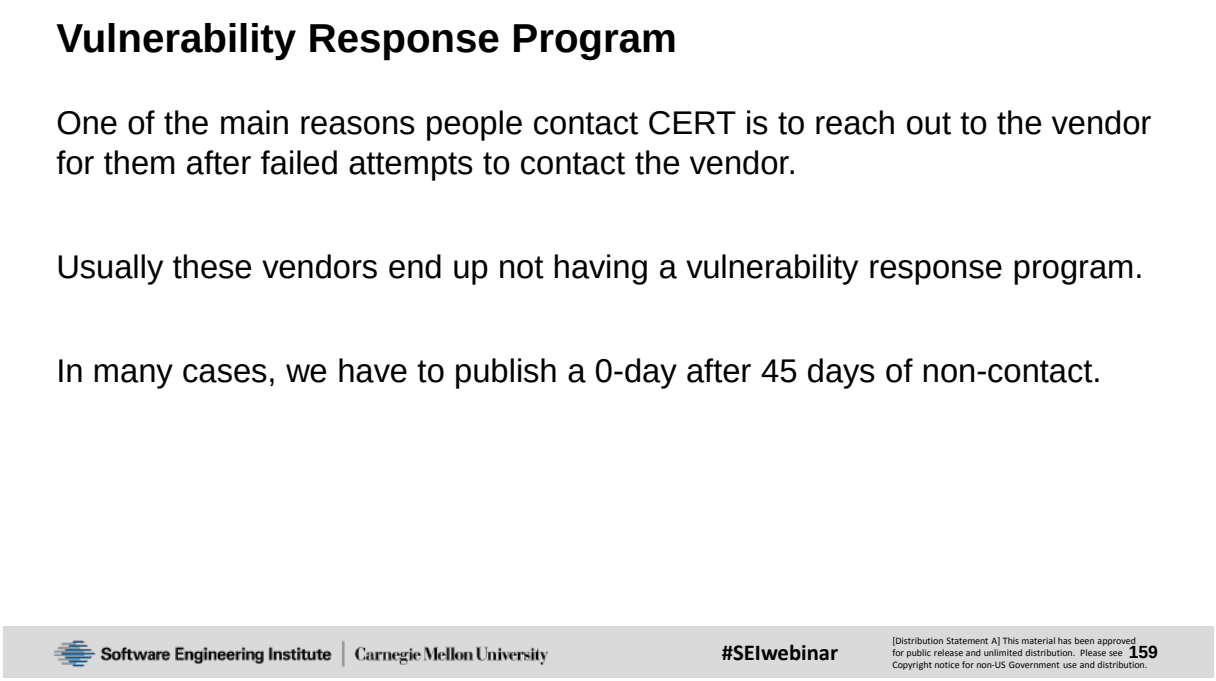
And finally, disclosure, best case is the vendor, the researcher, and CERT if we're involved, all sort of disclose the vulnerability at the same time.

Presenter: Okay, just to wrap up our polling question which was, "Does your organization have a vulnerability response program or product security incident response center?" Thirty-five percent yes, six percent in the process of building one, thirty-five percent handle it ad

hoc, and twenty-four percent not currently.

Presenter: Okay, that's I think about what we would expect to see. Like I said before, in traditional IT and larger software vendors, this process seems to be farther along and more mature. In industries and vendors who are newer to the process, it's not quite as mature yet.

Vulnerability Response Program

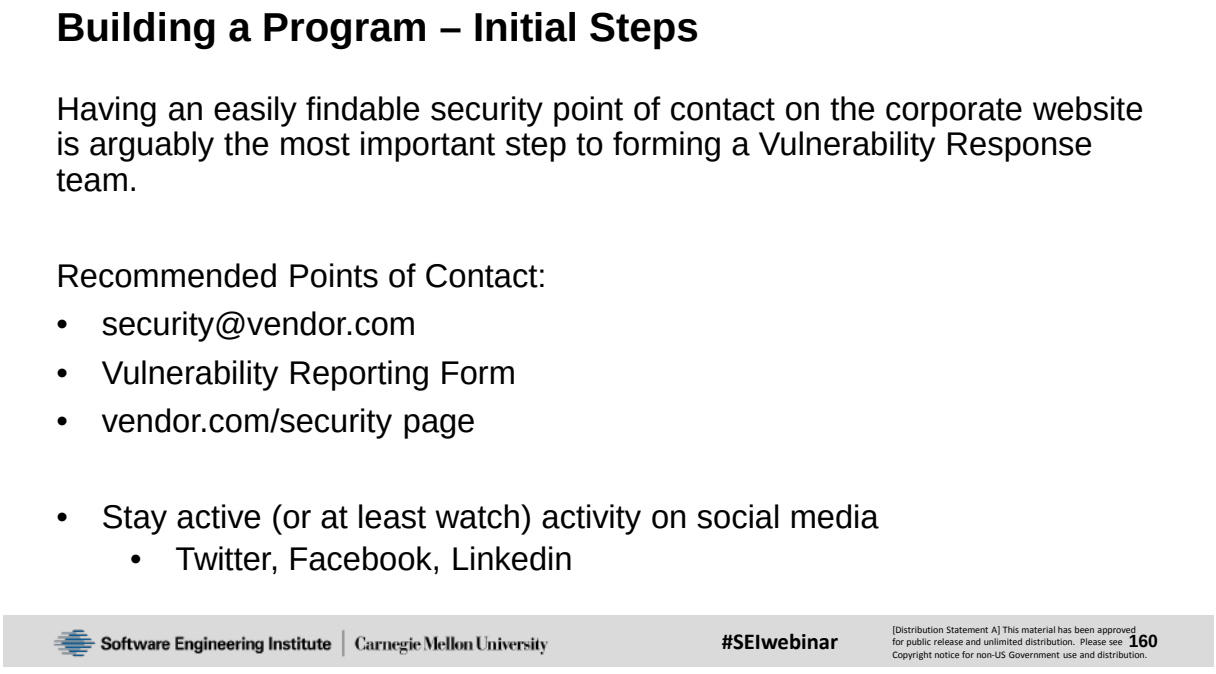


**159 So, as you're developing the security response program, the first issue is simply being able to take a report. One of the main reasons that people contact us at CERT Coordination Center is because they've tried to reach out to the vendor already to tell them that there's a security problem in their product. And they've been unable to get any answer. And sometimes, we

also are unable to get an answer from the vendor.

And what that usually means is vendor does not have any sort of vulnerability response program set up yet. And unfortunately, a lot of times what happens after this is that we end up publishing a zero-day vulnerability or the security researcher does because some time limit has expired waiting for the vendor. At CERT, our policy is that after forty-five days of no contact, we will go ahead and disclose the vulnerability. It should be noted that that's the minimum number of days. If vendors reply to us, and they're actually working on a fix, then we certainly are willing to negotiate that.

Building a Program – Initial Steps



**160 So, the very first steps that you should do is make it possible for

a reporter to actually report a security vulnerability to you. We've got a few recommended points of contact on the slides here. First of all, setting up an email alias security@yourvendor'sdomainname.com is the number one. Even if you would like to use another email address and publicize another email address for reporting security vulnerabilities, we highly recommend having a security@ email address because that's what everybody's going to try if they don't know anything else.

If you're a large manufacturer or large software vendor, you might want to put together your own vulnerability reporting form so the vulnerabilities can go directly into some sort of ticketing system or into your bug tracking system. And finally, a vendor.com/security page is another good standard where people can go to find all the current security vulnerabilities and whether the security vulnerabilities are applicable to them.

Presenter: Do you know how many people use third parties because aren't there some third party companies that provide this place to go?

Presenter: Sure. So, there are a couple of third party companies like HackerOne and Bugcrowd. And I believe that they have several hundred companies as customers. That's not the security researchers, but the people receiving the

vulnerabilities. So, that's certainly an option is to take in vulnerability reports through them. And they'll do a lot of the handling of the communication with the researcher. At this point, they're not tightly integrated with internal development tools. So, I think that's where people are still working to integrate those processes and one of the reasons you might choose to do this on your own. But yes, those outsourced options are certainly there now.

And then finally, staying active on social media like Twitter, Facebook, and LinkedIn, the people that monitor your accounts on those social media sites should know to be aware of the fact that people might try to report security vulnerabilities there.

Building a Program – Initial Steps

Building a Program – Initial Steps

- Identify a team member that can answer emails and has reach back into the engineering team
- Identify email address and advertise on public site
- Assign an internal vulnerability identifier
- Add workflow to Product support and PR groups for vulnerability handling
  - Product Support should forward vulnerability related communication to the designed POC.
  - PR should have a process for handling a vulnerability in your product

**161 So, as you're developing this

communication capability, the first thing you want to do is identify a team member that can answer emails, and importantly, has reach back into engineering. So, what you want to be careful of is that you don't-- if your marketing department handles incoming emails and your social media feeds, a low-level marketing person may not be able to reach into engineering easily to get the attention of somebody that can actually fix the problem. And so, you want to make sure you set up upfront that the person has some sort of communication channel into engineering to get the problem fixed.

You obviously want to identify an email address and advertise, as we mentioned. You probably want to assign an internal vulnerability identifier. Some companies use a whole separate tracking system just for vulnerabilities with their own identifiers. Some people just incorporate them directly into their bug tracker in general. There's not a really a preference between one or the other. But I think one concern with putting it directly into a bug tracker is that the security vulnerabilities that are known by third parties may not be drawn out as much to the attention of the developers as other things in the bug tracker.

Presenter: So, you mentioned having a communication path into development. Chris mentioned that FCA acted very engaging, did not threaten lawsuits or things of that

sort. What's been your experience with other companies if, in fact, this initial contact is not into development but into someplace else that gets routed into legal, gets routed into PR. Do you see that happening much? Or is that really the outlier at this point?

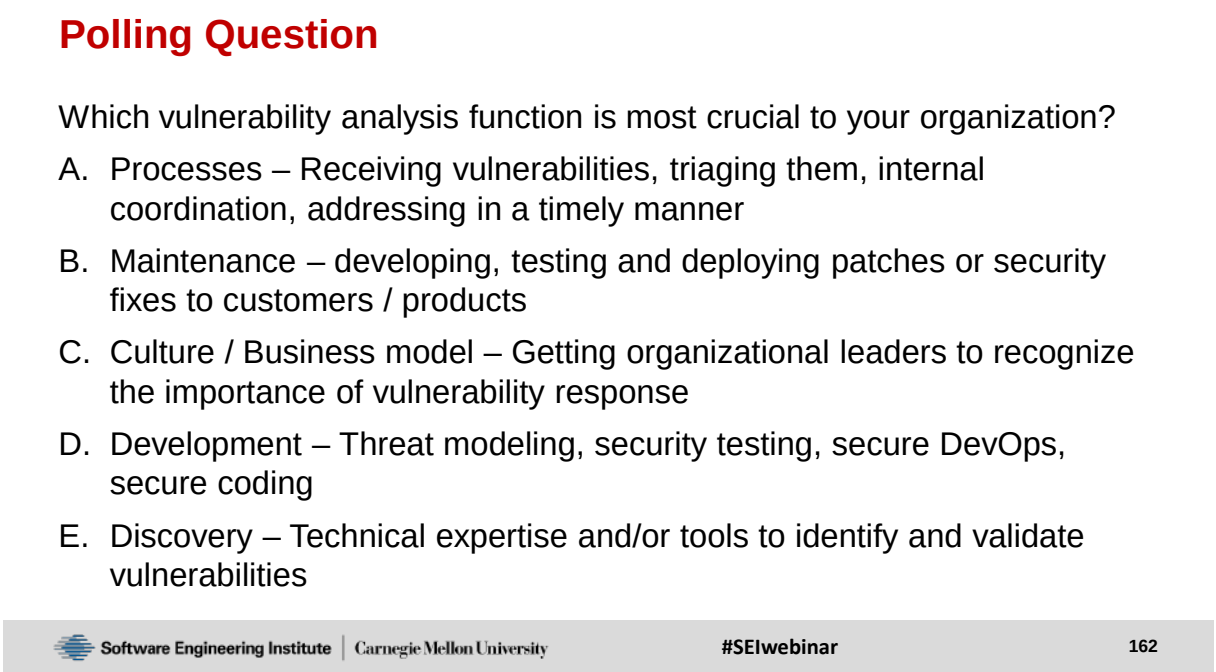
Presenter: No, I would say that we very frequently see that the first contact is outside of engineering. It might be product support. It might be PR. Rarely, I think is it legal. And those aren't necessarily bad things. They tend to find their way to the person that can make the decision or to engineering. It just introduces some delay. It's very rare to run across a vendor that is-- first talks about legal issues or potential legal actions before even looking at the vulnerability. And they definitely tend to be people that are-- vendors that don't have experience doing vulnerability response yet.

Presenter: Good.

Presenter: And then speaking of some of those external groups, another thing that you'll need to do is add workflow in for those groups. So, for example, product support, the support@ email address on your website, is probably going to eventually get security vulnerabilities reported to them. they need to know where to forward those, whether it's just forwarding them to the security@ website or some internal group that they need to forward it to. And then PR should have some kind of process for handling it also, even if

they just get notified about the vulnerabilities so they're aware ahead of time.

Polling Question



**162 Presenter: Okay, we're going to ask another polling question here. It will be our last one of the day. And what we'd like to know is which vulnerability analysis function is most crucial to your organization. You'll see a number of options on your screen now. So, while we'll give you a little bit to vote, we've got just a quick question for Dan here from the audience asking, "Are there concerns that hackers might use these vulnerability tools to explore security vulnerabilities?"

Presenter: So, that could potentially be an issue. And that kind of goes to a larger discussion that's been happening for quite a while now

which is, is the publicizing of vulnerability information actually advantageous to hackers because they find out about a vulnerability in your product which may or may not have been patched yet. And we-- our philosophy here at CERT has been for all these years that it's better to get the information out there and allow customers to make informed decisions about their risk and to take actions that they would choose to take. And I would say that that's true whether it's information about how to contact people, whether there's a compromise of a centralized system that you use to track vulnerabilities or whether it's the technical information of how to do the vulnerability itself.

Presenter: Okay, and just another quick one from Ellen, I'm pretty sure we'll have a quick answer to this one. But it says, "With your experiences, or both of you, in a program adhering to secure code development lifecycle, could an application ever be deemed a hundred percent secure?"

Presenter: No.

Presenter: No.

Presenter: Certainly, secure coding practices are going to reduce the number of vulnerabilities. But there will always be some.

Presenter: And I'll just wrap up the polling question which was, "What vulnerability analysis function is most crucial to your organization?" We had

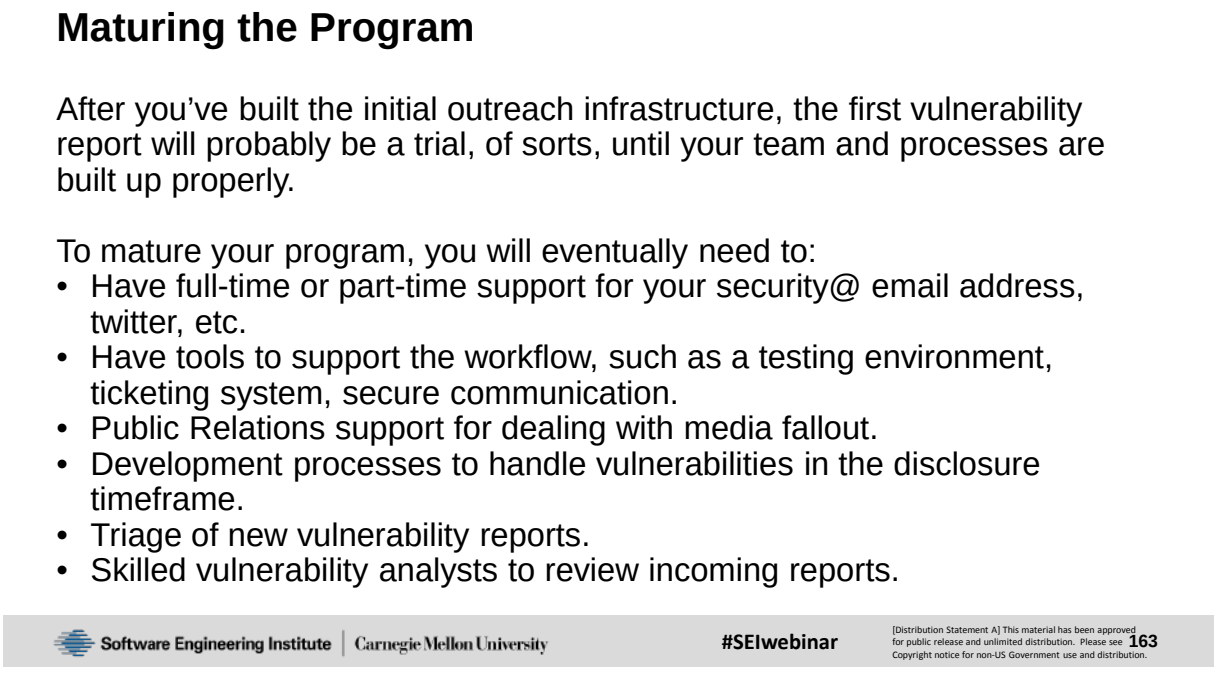
fifteen percent at processes, nine percent maintenance, fifteen percent culture or business model, forty-five percent-- or forty-six percent development, and fifteen percent discovery.

Presenter: Well, that's perfect because hopefully the rest of the seminars or the talks today have helped out in the development area. I suppose that may reflect the people who are signed up for the seminar. But I'm sure they've gotten a lot of use out of this today, then.

Presenter: All right, we'll turn it back to you, Dan.

Presenter: All right, thank you.

Maturing the Program



**163 So, I'm curious, you were talking in the various kinds of

disclosures. It was sort of internal and public. I'm wondering if there might be something in between using, I'll say, industry organizations, ISACs, in where if you are a provider in that particular industry-- I'm inventing something. I write software that's specific to the water plant industry. Communicating with them, those customers who use that software, rather than putting it out to the entire public as sort of an intermediate step along the way.

Presenter: Yeah, that's an option. I don't know how far along any of the ISACs are at doing that. I believe that the financial industry has a very mature ISAC. So, they may be exchanging that type of vulnerability information within themselves. I think in most industries, there's still concern about potentially allowing your competitors to see what vulnerabilities you have in your products. But I think that especially as we get into things that are safety critical like automobiles or other vehicles, there may be a good reason to have vulnerabilities that are disclosed within an industry trade group or within a subset of people and not necessarily published widely unless a fix can be pushed to the car or whatever automatically so that you're sure that there's going to be no safety impact from the vulnerability being disclosed. So, I would say that that is in its infancy, but probably going to grow quickly.

Presenter: So, a couple other steps about how to mature your program

once you've decided to go ahead and build one. Depending on the size of your organization, you may need somebody for part time just handling the security@ email address as well as any social media feeds. You'll probably need to build additional tools. We talked about integrating with your bug tracker or some other vulnerability tracking system. You're probably also going to need a testing environment so that you can do that validation piece where you decide how important a security vulnerability is and if it's really a vulnerability.

You'll probably need a secure communication set up ahead of time. Certainly, when we reach out to new vendors, one of the things that frequently takes some days of delay in the process is they might not have PGP keys or be too familiar with how to do secure communications. And as we saw earlier in my talk here, these vulnerabilities are worth money. So, it's worth protecting them very carefully with encryption or some other form of secure communication. We talked a little bit about public relations, being prepared to support issues, any kind of media fallout.

And then your development process is-- we kind of touched on this earlier, but when you're looking into how you do your DevOps and how you do your coding and engineering overall, having the fixing vulnerabilities process embedded into your development process is only going to help when the time comes to fix a vulnerability that's been

reported to you externally. If you have never planned for this and then all of a sudden you get a really large vulnerability, important vulnerability reported to you, then it's going to take a lot longer to get that fix prepared and tested and deployed than it would have if you had planned for it and put as part of your development lifecycle all along.

And then finally, the last two kind of point to the fact that you're going to eventually need some skilled vulnerability analysts who may or may not be part of the engineering team itself to do the validation and triage of new vulnerability reports. We typically see this in like a product-- a PCERT type of environment, possibly even in a security SOC or something like that. But it could, it depends on your organization whether you want to locate it in engineering or not.

Communication Plan

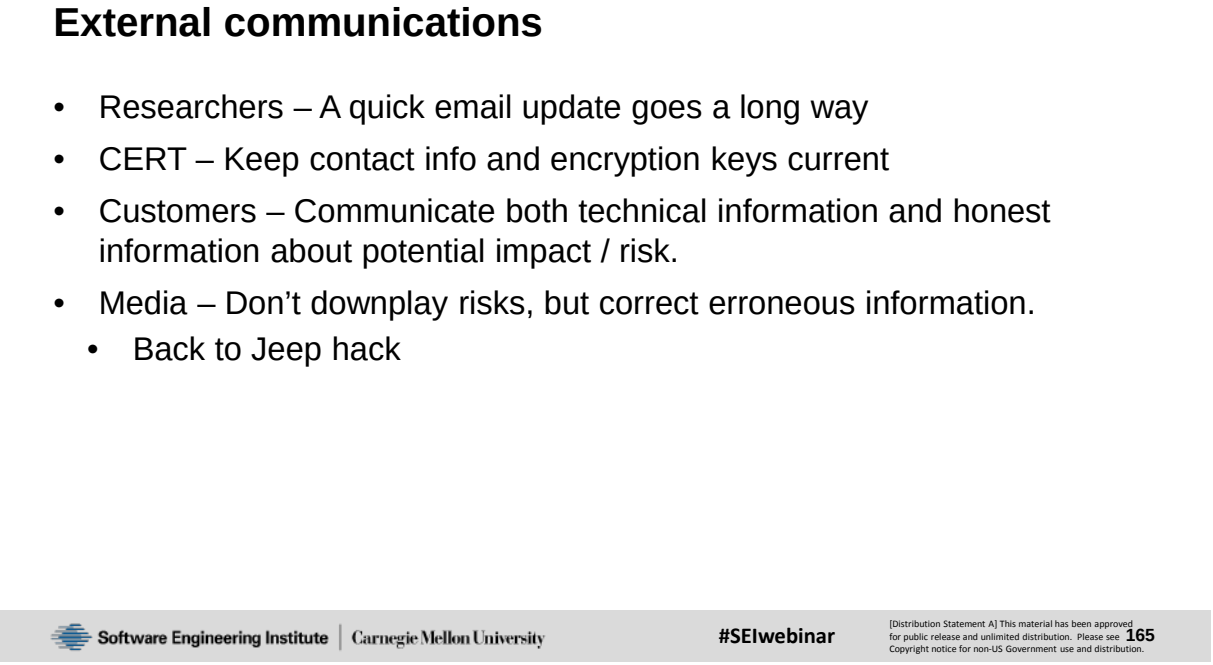
The vulnerability response communications plan should clearly identify points of contact, email aliases, and stakeholders.

The plan should identify how to handle situations such as:

- Active exploitation of the reported vulnerability is identified in the wild.
- The vulnerability is privately disclosed by two separate parties.
- The vulnerability has been made public before the agreed upon date.
- The vulnerability is present in more than just your own products.
- The vulnerability is more severe than the researcher suggests.

**164 One of the prime things that's going to come out of your early planning for a vulnerability response program is going to be a communication plan. And there's a bunch of possible scenarios up here. And these are the kinds of things that you should think about ahead of time because at some point, they'll probably all happen or some variation of them. So, it's worth formalizing your communication plan. You may go as far as doing sort of a tabletop scenario thing where you go through some of these with different stakeholders like PR and legal and things and talk about what happens in each of these different cases. Or you may just want to think through possible things that have happened and just put-- outline in a communication plan what you're planning to do in response to those. So, you're prepared.

External communications



**165 In your communications plan, you should also consider external communications. So, researchers, of their main complaints is that they put a security vulnerability into a support@ or security@ email address, and then simply never hear back from the vendor. And just sending them an email once a while saying, "Okay, we acknowledge that it's a real vulnerability. Okay, we're working on a patch. This is the release date," goes a long to way make sure that they are working with you rather than against you. Working with CERT, it never hurts to already have your contact information, encryption keys, with us so that we can reach out to you really quickly in case of any high profile vulnerabilities.

Your customers, you want to make sure you communicate to them both

the technical information about the vulnerability and what they can do to mitigate it, but also some honest information about the potential impact. If it's going to be a major impact to them, then you want to tell them so that they can go ahead and start planning accordingly.

And then the media, our recommendation is you don't want to downplay the risks of problems. But then you also want to make sure that you correct erroneous information. So, the Jeep hack for example, FCA already had a patch ready for it when Chris and Charlie talked about the problem. And Sprint blocked that particular attack path within a few days, as Chris said earlier. But there were press reports coming out for days, weeks, probably months saying your Jeep can be hacked. And so, FCA-- I'm not too familiar with their internal processes, of course, but I imagine that they tried to at least correct the media accounts that said that this was still an issue after some of the main problems had been mitigated.

Okay, so there's sort of a balancing act there for a vendor. You don't want to downplay things. You also don't want to let it be sort of blown out of proportion.

Resources

ISO Standards

29147 Vulnerability Disclosure

- ISO/IEC 29147:2014 gives guidelines for the disclosure of potential vulnerabilities in products and online services. It details the methods a vendor should use to address issues related to vulnerability disclosure.

30111 Vulnerability Handling Processes

- ISO/IEC 30111:2013 gives guidelines for how to process and resolve potential vulnerability information in a product or online service.

**166 So, there are a couple of ISO standards that you can look up as resources. And I'm not going to ready through these, but I'll leave them up there so that people can jot them down and look them up if they'd like. And while you're doing that, we're getting a little low on time. But I'm happy to take questions.

Presenter: Great, yeah. We'll get one from Derrick asking are compiler vendors falling short. Could they do a better job in finding code security issues? And why don't they, if they are falling short?

Presenter: Go ahead.

Presenter: Sure. So, this goes back to the general tool question. And like all companies that build things, they

make choices among all the different kinds of features they could add to their products. And one of those features are additional checking for security rules. In fairness, if you look at how much of the market works for compilers, the first focus is on speed. And so, the compiler generates the fastest code usually is the one that people focus on. And--

Q&A

Contact Information:
Dan Klinedinst
djklinedinst@cert.org

Public Vulnerability Information:
www.kb.cert.org/vuls

Software Engineering Institute | Carnegie Mellon

**167 They're like every other product company. In the open source communities, things like Clang, again, it depends on who's involved and what the interest is of those open source contributors. As it turns out, we are one of the contributors. We're committed to Clang. And we do add some checkers. But we have a limited amount of resources. And there's a lot of people who are working on this.

Presenter: And then just one last one. I know your time is up at the end. But it came in during this section, Robert asking have you go any book recommendations on the subject. I don't know if that meant vulnerability, disclosure itself, or just secure coding in general. So, maybe I'll leave it to both of you there.

Presenter: Well, I can quickly answer that I do not know of any books on the vulnerability and disclosure process. Although, several people have expressed interest.

Presenter: Hence, the training-- Mark, how about you? I know there's a number of secure coding books. Can you rattle off a couple?

Presenter: So, as Robert may be well aware, there are some books on secure coding that have come out. You can go to Amazon and basically search for them and you'll find them. In addition, we have a wiki here at CERT in which all the secure coding rules are also made available to anyone who wants to connect to that wiki at least for the ones that we have developed. Obviously, ones that are developed by other organizations, there are other organizations looking at this. I don't know how many of them have actually published ones on security. Things like Misra have been focused on safety, for example.

Presenter: Okay, and thank you for your presentation. So, we're going to go back to Mark with just wrapping up today's events, all the things we

learned and some possible next
steps. So, Mark, back to you for the summary today.

SEI WEBINAR SERIES | Keeping you informed of the latest solutions