Vulnerabilities

At a glance

Description	A look into software vulnerabilities, whether we are making any progress in addressing them and ways to improve.
Contributors	Kenna Security (formerly Risk I/O) collaborated with us again to leverage their vulnerability and exploitation data. We also utilized vulnerability scan data provided by Beyond Trust, Qualys and Tripwire in support of this section.
Key findings	Older vulnerabilities are still heavily targeted; a methodical patch approach that emphasizes consistency and coverage is more important than expedient patching.

New vulnerabilities come out every day.

Methodology
The visualizations and statements regarding rates of exploitation in this section are underpinned by vulnerability exploitation data provided by Kenna Security. This dataset spans millions of successful real-world exploitations, and is derived from hunting down exploitation signatures in security information and event management (SIEM) logs and correlating those with vulnerability scan data to find pairings that would be indicative of a successful exploitation.

The tortoise and the hare
Vulnerability management has been a Sisyphean endeavor for decades. Attacks come in millions, exploits are automated and every enterprise is subject to the wrath of the quick-to-catch-on hacker. What's worse, new vulnerabilities come out every day. Since the first DBIR, we've been advocating the turtle's approach to vulnerability management (slow and steady wins the race).

This year we revisit this data to see whether the trends hold, but in typical DBIR fashion, we dig a little deeper, to look at not just how attackers are interacting with vulnerabilities (exploitation), but also how well and how fast enterprises are executing remediation. If we can measure both of these routinely, then we can provide much-needed answers about how the tortoise won the race—and so learn how to close the gap between attackers and enterprises.

Slow and steady – but how slow?
This year we take a different approach to measuring the time from publication to exploitation. Figure 10 is a box plot, which plots the time between publication and the first observed successful exploit by vendors.[6] We can see that Adobe vulnerabilities are exploited quickly, while Mozilla vulnerabilities take much longer to exploit after disclosure. Half of all exploitations happen between 10 and 100 days after the vulnerability is published, with the median around 30 days. This provides us with some general guidelines on which software vulnerabilities to prioritize along with some guidance on time-to-patch targets.

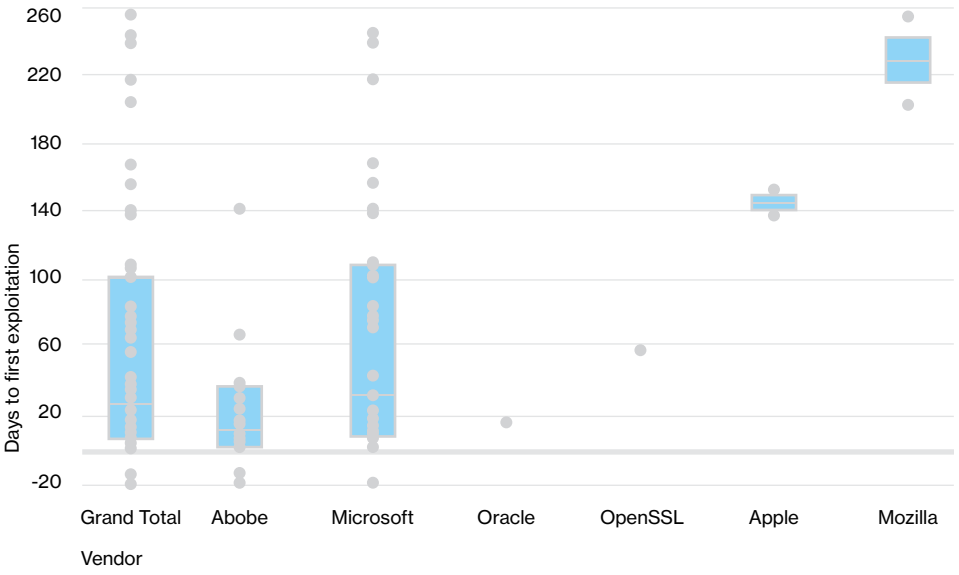

Figure 10.
Time to first-known exploitation by vulnerability category.

Treading water
Figure 11 shows the number of vulnerabilities opened each week minus the number of vulnerabilities (aka "vulns") closed, scaled by the number of assets in the dataset during each week of 2015. When the line is above zero, it means that more vulns are being opened than closed (new vulns disclosed, more

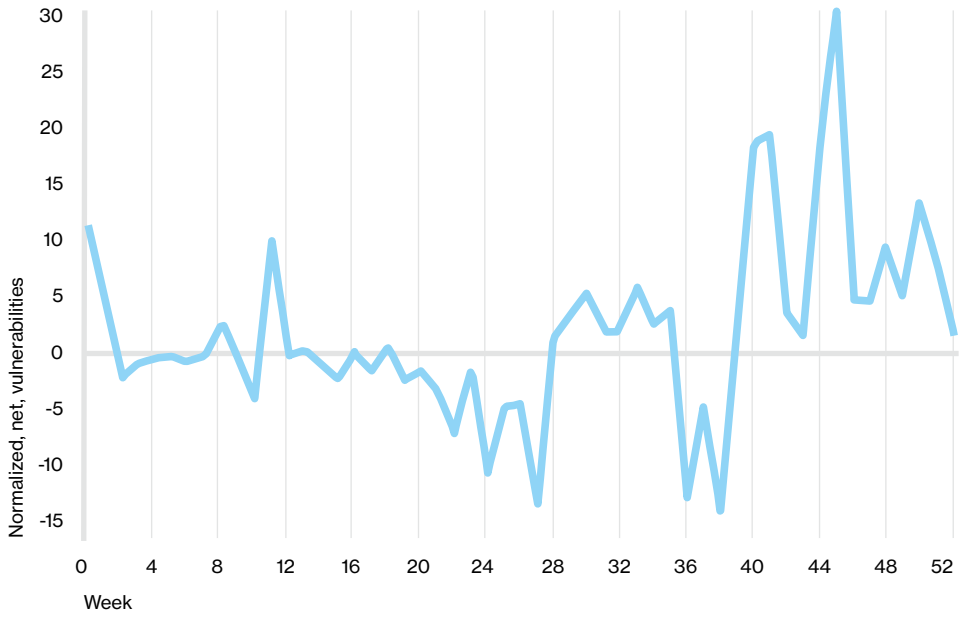

Figure 11.
Delta of number of vulnerabilities opened each week and number closed.

6 The blue boxes in Figure 10 represent 50% of the values for a given category and the gray line within the box is the median value. The dots represent individual values.

machines entering the environment, new software installed). When it's below zero, remediation efforts are driving down vulnerability counts faster than new vulns are entering the enterprise.

Basically, we confirmed across multiple datasets that we are treading water—we aren't sinking in new vulnerabilities, but we're also not swimming to the land of instantaneous remediation and vuln-free assets. However, all that patching is for naught if we're not patching the right things. If we're going to tread, let's tread wisely.

All that patching is for naught if we're not patching the right things.

What should we mitigate? Hacker economics.
So what are the right things? The 2015 DBIR gave us an idea and since then, not much has changed.

Revisiting last year's trends, we find that the two golden rules of vulnerabilities still hold.

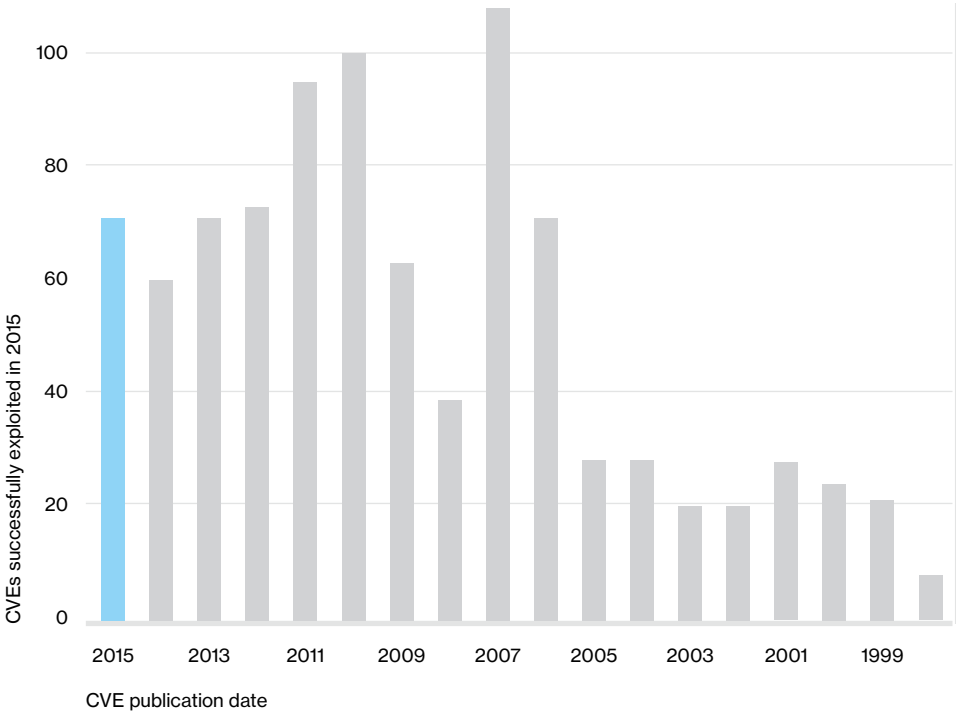

Figure 12.
Count of CVEs exploited in 2015 by CVE publication date.

First, Figure 12 arranges CVEs according to publication year and gives a count of CVEs for each year. While 2015 was no chump when it came to successfully exploited CVEs, the tally of really old CVEs which still get exploited in 2015 suggests that the oldies are still goodies. Hackers use what works and what works doesn't seem to change all that often.[7] Secondly, attackers automate certain weaponized vulnerabilities and spray and pray them across the internet, sometimes yielding incredible success. The distribution is very similar to last year, with the top 10 vulnerabilities accounting for 85% of successful exploit traffic.[8] While being aware of and fixing these mega-vulns is a solid first step, don't forget that the other 15% consists of over 900 CVEs, which are also being actively exploited in the wild.

7 Astute and frequent readers of the DBIR will notice one more gem in this chart—last year, the numbers of published CVEs exploited were lower across the board—and this year, we have more and better data. Those newly exploited CVEs however, are mostly—and consistently—older than one year.

8 CVE-2001-0876, CVE-2011-0877, CVE-2002-0953, CVE-2001-0680, CVE-2012-1054, CVE-2015-0204, CVE-2015-1637, CVE-2003-0818, CVE-2002-0126, CVE-1999-1058.

Can't solve everything
In Figure 13, we see that during 2015, vulnerabilities published in 2015 and 2014 were being patched. After that though, the vulnerabilities begin to drop off and really hit a steady state. This gets at a core and often ignored vulnerability management constraint—sometimes you just can't fix a vulnerability—be it because of a business process, a lack of a patch, or incompatibilities. At that point, for whatever reason, you may have to live with those residual vulnerabilities. It's important to realize that mitigation is often just as useful as remediation—and sometimes it's your only option.

Mitigation is often just as useful as remediation—and sometimes your only option.

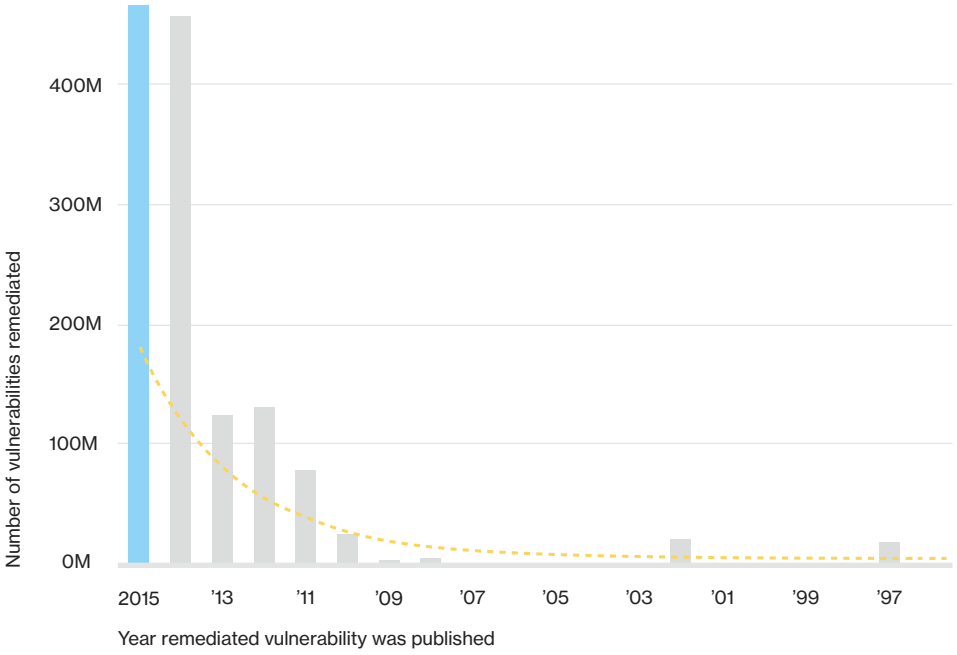

Figure 13.
Closure rate of CVEs by CVE publication date.

Recommended controls

Knowledge is power.
Establish a process for vulnerability remediation that targets vulnerabilities which attackers are exploiting in the wild, followed by vulnerabilities with known exploits or proof-of-concept code.

Have a Plan B.
If you have a system that cannot be patched or receive the latest-and-greatest software update, identify it, and apply other risk mitigations in the form of configuration changes or isolation. Discuss a plan on how the device(s) could be replaced without causing severe business disruption.

At your service
Vulnerability scanning is also useful in identifying new devices and new services. Review scan-to-scan changes as another control to identify unknown devices and deviations from standard configurations.