SAGE First Inc.

The History of SAGESecured for Android OS

SAGEFirst™
<process-based solutions>

History of SAGESecured…
SAGESecured began as the brainchild of a Geek's geek – the type of person who writes their own operating system and codes in Assembler for recreation.

It all started in Amarillo, Texas, the Internet Security capitol of the universe since the late 1990s. **:~)** Once a week, he'd comment that the virus or worm of the week could be easily prevented. After several weeks, he was challenged to prove that such "preventable" attacks could indeed be avoided.

This was the last days of the DOS operating system and he had a demo with DR DOS. It included an infected file from a word processing application. When the file was opened it was immediately denied access and no arbitrary code was surreptitiously loaded and executed. With the introduction of Windows by Microsoft, the days for both DR DOS and MS DOS were numbered. That was a shame because both operating systems could be readily modified to demonstrate the merits of the change prevention protocol that was imagined. We missed our chance to save the (cyber) world at that time.

With the advent of Windows, everyone seemed less concerned with security and more just fascinated by the shiny new toy that Microsoft had shared with the world. Because Windows was proprietary, there was no way to introduce a security protocol that can prevent all worms, viruses, Trojan horses and other malware that is surreptitiously loaded and executed and that conventional best practices can only address after a cyber-autopsy. The next virus, etc. is always going to have free rein until its profile can be distributed in an update. Polymorphic exploits basically render anti-virus software a moot point.

About that time a college student from Finland gave the world a non-proprietary operating system called Linux. Linus Torvalds gave the world an updated version of Minix which was a Unix derivative. That was the first widely distributed open source operating system and would be used by us to demonstrate the robust nature of the change prevention protocol we propose be used to permanently eliminate worms, viruses, Trojan horses, etc. With the arrival of an open source operating system that was available for people to play with, we were able to create a secure web-server appliance by retrofitting our solution into the open source Linux. The resulting appliance included an email server, an FTP server and a website hosting server.

About that time there were several high profile web hacking incidents that were making headlines. An Albuquerque ISP was so severely hacked that the ISP was offline for several days while the servers involved were rebuilt. It turns out that a raging feud between a site hosted by the ISP and several black hatted hackers caused the black hats to take down the entire ISP when the targeted website was too difficult to subvert. It is indeed a delicious irony that Albuquerque should be the site of such an exploit. Microsoft was born in Albuquerque and had only left for Seattle a short time before the Route 66 exploit.

A second exploit at that time was of Security News Portal. The website was also the target of black hatted hackers who targeted another hosting ISP. Security News Portal publically explained that the ISP was going to shut the site it was hosting because the site was ostensibly an attractive nuisance. As a result we reached out and offered to host the soon-to-be-abandoned website.

So we took two high profile targets that had previously been destroyed and used these "toxic" sites as demonstration projects for prototypes of a server that incorporated the protocol we propose. It could be used for Android users who want to get off the security treadmill and use an operating system that addresses and solves the problem instead of dealing with the symptoms. What we propose for Android has been time-tested with servers. For more on the matter please review: http://www.securitynewsportal.com/faq.shtml and http://www.forbes.com/forbes/1998/1116/6211132a.html.

Around this time there was an article we read about a penetration testing group out of Sandia National Labs in Albuquerque. After a brief back and forth we gave them a seminar on the technology, all the source code and access to the developers. Then we gave them a server with a challenge to either tell us the content of a particular file or to change the content of a particular file. After several months they told us they did not accomplish the challenge but did report some problems with the Linux operating system that were not a reflection on the change prevention/change control protocol we recommend people adopt. Finally they pointed out some hypothetical problems with Linux that did not result from the design of the security model we presented them. The Army's Communications Electronics Command at Ft. Monmouth, NJ (CECOM) also inspected the protocol. This consisted of an evaluation of an implementation of our secure web-server appliance on a mobile Compaq Ipaq server. Again they did not subvert the security model protecting the contents on the Ipaq from malicious third party fiddling. After the results with Sandia National Labs and CECOM…. problems involving several intractable circumstances were solved using our secure web-server appliance.

The National Library of Medicine at the National Institutes of Health had a public facing website that was being looted and plundered on a daily basis. Once the site was hosted on a server invoking the protocol we advised to be used, the website was available all day every day. The public was able to get access to needed information without the bother of trolls preventing access.

Meanwhile, the Incident Response Center of the Nuclear Regulatory Commission (NRC) on the other hand wanted to collect reports from the 100 + nuclear power plants that it supervises. The NRC was transitioning from a report collecting system that incorporated faxes, emails, snail mail and even phoned in reports to an Internet based system that consolidated the reporting and sped up the collection of those reports for processing. By implementing our change prevention/change control protocol, they were able to keep unauthorized users away from confidential information.

An example from the private sector…

A commercial entity in the business of overnight delivery of letters and parcels has one of the larger fleets of planes in the free world. That means that they have lots of pilots, co-pilots and other personnel who are constantly updating their time in particular types of aircraft. In addition there was just as large if not larger a pool of people wanting to apply for work with this company who also needed to maintain their log books. Keeping up with all that information was becoming a daunting task especially so when pilots could be in any time zone when completing a leg and they wanted to update their logs.

When this company wanted to simplify their credentialing process they put it on the Internet on a secure web-server appliance that was designed specifically to keep unauthorized prying eyes out of the very expensive database of highly skilled and trained individuals. In addition the company wanted to keep the information segregated from the accounting system so this credentialing system was put in front of the firewall directly exposed to the cold cruel Internet in order to allow the personnel easier access to the logs that they updated when they completed a trip.

These examples of real problems with real solutions demonstrate the implementation of our protocol in custom built solutions. After these installs occurred, these various problems were solved. Their servers have performed without a hitch.

At the time we could only demonstrate the merits of our security on Linux which is not proprietary but never was really adopted to such a level as to attract the attention of crackers as Microsoft attracted attention.

Now with the advent and market acceptance of Android, off the shelf implementation of this solution is available to a broader audience. With the completion of the porting to Android of the same security model, everyday users can surf the internet and know that they are not going to get drive-by downloads of extraneous applications that some complete stranger might try to run.

In the period including the early 2000's security took a decidedly backseat to bells and whistles that Windows was rolling out. Then came the famous "Trustworthy Computing" memo from Bill Gates to the crew at Microsoft in mid-January, 2002. For the next year or so the market expected some results from that effort to recognize the need to build more robust security into operating systems and applications.

More than a decade later the results of the memo have not reduced the need for patch Tuesday (the second Tuesday of each month) or lessened the reliance on anti-virus. Remember anti-virus was a cottage industry in the early 2000's that was supposed to be a short term bridge between the then existing problem-plagued operating systems and the systems that incorporated that promised "Trustworthy Computing".   Since then antivirus has grown like the eggplant that ate Chicago and is about as useful.

After it became clear that proprietary operating systems were going to rely on the End User License Agreement and not address the underlying cause of the ongoing problem which surreptitiously loads and executes arbitrary code we decided to hunker down and wait.

That wait was finally ended with the introduction of Android in November 2007 with the beta release and the September 2008 commercial release of the first version of Android. With the roll out of Android, there is now a well-received operating system where we can demonstrate our protocol and the market can decide how much insecurity they choose to endure.

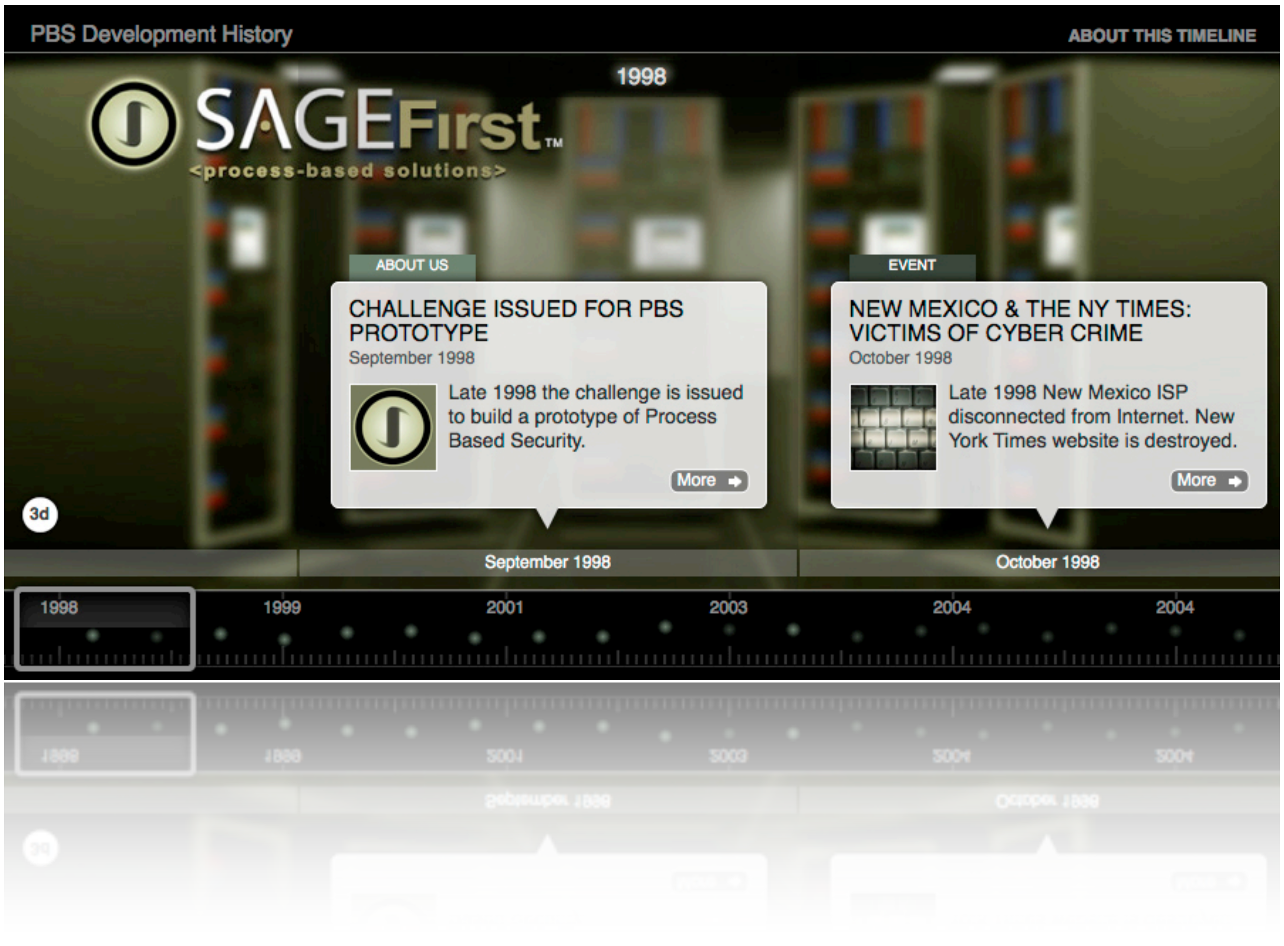
In the wake of the advent and market acceptance of Android, off the shelf implementation of this solution is available to an initially skeptical but ultimately grateful public. With completing the port to Android of the same security model that has been proven in servers, everyday Android users can surf the internet with confidence knowing that they will not be subjected to drive-by downloads of extraneous applications that some malicious coder might desire to run.

Once Android cracked 50% of smartphone operating systems we decided that it was for real. We began the port in the third quarter of 2011. At the end of the 4th quarter of 2012 Android held a greater than 70% market share for smartphones. It will probably exceed 500 million installations for smartphones in 2013.

No more worry about keystroke loggers, spyware, ransomware or Trojan horses bent on stealing a person's banking credentials. There is no telling what sort of exploit shenanigans are already in store for the Google Glass product which is still in beta. Today, Android allows for more rank and file computer users (desktop, laptops, notebooks, smartphones etc.) to benefit from the enhanced control of what does and does not run on any system. It's that simple, and that revolutionary.

PBS DEVELOPMENT HISTORY TIMELINE

Explore our interactive PBS Development History Timeline by clicking here or on the image below.



CHALLENGE ISSUED FOR PBS PROTOTYPE
September 1998
Late 1998 the challenge is issued to build a prototype of Process Based Security.

NEW MEXICO & THE NY TIMES: VICTIMS OF CYBER CRIME
October 1998
Late 1998 New Mexico ISP disconnected from Internet. New York Times website is destroyed.