

WHITE PAPER

Defending in Depth

Daniel E. Geer, Jr., ScD
Vice President, Chief Scientist
Verdasys, Inc.
October 2005

VERDASYS[™]

Data Security at the Point of Use

If ever there was a place where working smarter made more sense than working harder, then digital security is that place.

1. What are We Going to Talk About?

Ladies and Gentlemen, we are going to talk about a nasty world, what our strategies and choices are for dealing with a nasty world, how we can make progress by borrowing from other fields, why we should now remember the wisdom of our elders, and why versatility is the key to winning an(y) arms race. To some degree, this is conjectural. To some degree, it is very good that this is conjectural as one would rather not experience every bad thing just to confirm that they are bad, just as you should not have to buy one of every conceivable security tool just to prove to yourself that they are mutually redundant yet not collectively exhaustive. If ever there was a place where working smarter made more sense than working harder, then digital security is that place. We believe that we've hit on a solution to this set of equations. We will speak of general truths for as far into this paper as we can, and then we'll tell you when we are switching over to our particular innovations which we want you to accept as truth and progress. You be the judge, but here's the evidence.

2. The State of the World

Obviously, all commercially viable software systems of any appreciable value are complex. Obviously, that complexity both enables us and defeats us. Because we are security people, we are going to focus on mitigating the defeats that complexity brings us.

The state of the world is that there are attacks of all sorts, nearly too many to enumerate. In fact, there are too many to enumerate though we will have to return to that later. Some attacks are against integrity, some against confidentiality, some against availability, and some that have more than one target. Some of these attacks are directed by hostile and sentient parties on the outside of the firm, some on the inside, and some are the result of stupid mistakes. In fact, there are lots of executives that would readily suggest that two-thirds of all security breaches are traceable to user error. For the purpose of this discussion, we think that one gets a powerful result if you can ignore motive (and thus class spectacular stupidity with spectacular hostility if they have the same net effect).

So, one thing to note about attacks is that the skill necessary to create them is in ready supply (and increasing at the high end) but the skill to perform them is trending downward (as attacks are packaged in even more convenient tools any idiot can use). Hence one conclusion: Exacting a "skill tax" as a way of lowering the net availability of attacks is going to be a slow, painful, and not terribly effective process.

Another thing to notice about attacks is that they escape the surly bonds of earth – location is irrelevant. Thinking like an infectious disease expert, you are rather likely to catch cold if you shake one million hands an hour. In the same way, everyone on the Internet is approximately equidistant which, to be more blunt about it, is a way of saying that every sociopath is your next door neighbor. Hence a second conclusion: Hope (that you will not be found) is not a strategy.

One more thing to notice about attacks is that they are professionalizing. Not overnight, and the background radiation pressure of a jillion misanthropes running scripts cannot be overlooked, but as of now the growth sector to worry about is the professional. Money is the object, and money does eventually work. It used to be you had to work hard to steal a million dollars all in one go and there would be bullets involved. If you can steal one dollar a million times, you have the same net gain and there won't be bullets – there won't even be much of a fuss (for a dollar no police department can afford to care). So, another conclusion: The opponent is after data and the opponent is automated; this is one Jones with whom you must keep up.

A final observation about attacks is that they don't wait around. The delay time between when the knowledge of a method of attack hits the airwaves and when automated attacks based on that method appear is now measured in hours. In fact, and this is the chilling part, to prepare to repulse such an attack you know is coming can take longer than the time between when you learn about the risk and when the attack appears. This may induce you to install patches without testing, thus elevating your risk of reliability failure. So, a final conclusion to go with the final observation: Whatever you do, it absolutely, positively, without-a-doubt has to work without imposing last minute fire drills. Those fire drills cost money, and they are what the old aphorism "Haste makes waste" was talking about.

3. Strategies and Choices

So, what are your strategy choices? Not many, but each is a big bet in the same sense that the Maginot Line was a big bet. First, you can harden your systems. Principally, this has a security benefit by way of complexity reduction. Complexity reduction is a wonderful thing, but it is not a panacea and it does have a limit. Mostly this means "Don't run programs you do not provably need" and that is, indeed, good advice. (You can even sell that on performance grounds.) But what if it is the remaining applications that are the rightful root of your worry? Lots of statistics show that these days breaking large applications (that have their own complexity problems) is easier than breaking infrastructure and, ipso facto, that is where the attacks tend to go. One must conclude that complexity reduction can go on as long as you have switches that you can turn off, and no further.

Second, you can stock up on weaponry. For probing attacks you can invest in firewalls, both at the corporate perimeter and on the desktop. Of course, you will have to work hard to make all those firewalls work together for the common good. That exchanges a security functionality problem for a configuration synchronization problem, so you can buy the services of one of the companies that analyzes firewall configurations, too.

For impersonation attacks, you can invest in authentication including the newly trendy two factor authentication. Of course, authentication is just the raw material for authorization, so let's refer to both as access control. Access control is a problem the work factor of which is proportional to the product of the numbers

It used to be you had to work hard to steal a million dollars all in one go and there would be bullets involved.

of users times the number of services, and ultimately anything that scales faster than linear (like a product) is a cost that cannot be indefinitely maintained. That isn't to say throw authentication, authorization, and access control away, just that you shouldn't expect them to do ever more complex and/or time-sensitive things or you will soon cost yourself more than you can afford.

For mobile code attacks, by which we mean the sum of worms, viruses, trojan horses embedded in other things, and even the misuse of remote procedure calls (SOAP and/or .NET), you can invest in explicit countermeasures like anti-virus systems. Of course, with a new Microsoft Windows virus every four hours around the clock, to keep your anti-virus system up to date you are going to have to update it every four hours around the clock (while hoping that the anti-virus vendor has something for you on time, in time, every time). This is not going to get any easier, and frankly we cannot see how it is that the anti-virus vendors can continue to make money with their present pricing structure. That is not likely good news for you, the end user.

For human opponents who are close to getting in, there are intrusion detection systems and even intrusion prevention systems. As with anti-virus, either they miss things or they require constant update with the latest signatures. If they do their work based on anomaly detection, then you get to choose the tradeoffs between false-negative rates and false-positive rates since it is rarely if ever possible to minimize both with a single test harness, a fact that is as true in medicine as it is in digital information security. Of course, you can tune down the false negatives and then buy yet another product to read your logs for you – a two-stage screening process that exactly parallels, say, the problem of detecting smuggling of weapons onto airplanes.

For human opponents who are already in – maybe because they work for you – you have to have some sort of enforceable policy and enough surveillance to go with it. This is where the classic requirements for a security product, viz., that it be undetectable and inescapable, come most to the fore. But before you go out to buy yet another special purpose product to go after insiders, products like robots that read e-mails looking for Social Security Numbers, remember that the complexity of your environment in sum means that the subject who knows he or she is under surveillance has only to think of an alternate way of, say, getting the data out the door. When it comes to insiders and the digital age, who is it that has to be perfect (it used to be the criminal) has reversed (now it has to be the defender). That changes everything.

4. Versatility is the Key

Every one of the digital information risks (problems) we've mentioned, and we have not mentioned them all, can be adequately addressed by onesies or twosies. For every problem that isn't rare, someone is already selling a solution targeted at

When it comes to insiders and the digital age, who is it that has to be perfect (it used to be the criminal) has reversed (now it has to be the defender). That changes everything.

Defense in Depth (DiD) is both a principle and a strategy.

that problem. Buy one of each? Of course you can. That would be little different than having a medicine cabinet with a separate antibiotic for every germ that has ever been identified. As soon as one of those germs shows up you take the perfect pill. All well and good, but you'll break the bank if you try this, and, unless you are really way, way ahead of the game, this is precisely what you are trying to do today. With the help of the ever helpful vendors, of course.

So, you have to, as Steve Jobs would say "Think Different." Actually, you have to remember things we used to know and, in our irrational exuberance for all things digital, happily and handily forgot: You have to have a fall back. When A fails for whatever reason (envy, sloth, gluttony, wrath, pride, lust, or greed), you better hope that there is a B standing behind A ready to back A up. Old western movies had this ("Cover me!") and, less flippantly and more to the point, serious computer security people had this only they called it by its righteous name: Defense in Depth.

Defense in Depth (DiD) is both a principle and a strategy. It is a principle in that it says no protection mechanism should be the last defense and it is a strategy in that it allows a rational array of defenses each one of which is insufficient but which collectively are sufficient. DiD is thus a well established military theme and a venerable if not well established digital information risk theme. It is our job today to convince you that this is a theme, an idea, whose time has come.

To pick literally at random, think about anti-virus. Its job is to recognize malware at zero latency and zero false positives. Strictly speaking, this is impossible so there must be some false negatives or some false positives or some delay. Delay will tend to be a permanent cost, so let's assume that delay is unacceptable. Thus, the anti-virus program must either have no false positives at the price of some false negatives or no false negatives at the price of some false positives. As it happens, the need to get real work done not only demands zero delay it also demands the minimization of false positives as taking priority over the minimization of false negatives. In other words, there will be occasional virus infections that pass the inspection of the anti-virus system. This is not a knock on the diligence of anti-virus but rather a cold assessment of what must be.

So, suppose once in a while it sort of has to be that anti-virus will pass a virus through unhindered. Then what? Here's what then: There are only two things that will then happen that matter in any way at all. Either some bit of data will be altered (and we are including both data in the colloquial sense and data in the sense of program files), or some bit of data will be sent to a distant location that is unwanted (embarrassing, dangerous, expensive, ...). That's it – modification of that which should not be modified or revelation of that which should not be revealed. Sure, there are lots of variations on these themes, but it is the same melody underneath.

Host-based protections are just about the only workable idea at the level of attack we have today.

What would DiD be for such a setting? Simple, actually: DiD would simply be to say what data can be altered and what cannot plus what data can be shipped out and what cannot. DiD might, we suppose, be thought of as “symptomatic relief” in that it didn’t actually cure the disease (virus infection) but rather it prevented the disease from having any untoward effects. There’s nothing wrong with symptomatic relief if it is well chosen. In the medical world as we write this, a Harvard researcher is homing in on a way to prevent tumors from getting bigger than the head of a pin. He doesn’t kill tumors he just stunts them. You still have them, but they never amount to anything. This is symptomatic relief, it cures nothing, but it is hard to argue that it is not good enough. That is the spirit of DiD.

Firewalls, like anti-virus, try to identify malware at wire speed. They may be downstream from malware and hence are blocking attacks, per se, or they may be the target of the attack themselves in that the malware may be attempting to pass through them. Either way, this is almost exactly like anti-virus in that it offers some awkward tradeoffs between false negatives, delay factors, and false positives. As before, delay comes off the table first. As before, the way in which firewalls fail leads to a need for defense in depth. It is not possible for it to be otherwise unless and until our opponents thoughtfully include warning labels with their attacks. And, in agreement with the Gartner Group, intrusion detection systems are just firewalls with extra features.

While we are doing this inventory, it is good to remember that host-based protections are just about the only workable idea at the level of attack we have today and with the dispersal of valuable data and programs that we do for all sorts of wonderful business reasons. To a large degree, the dual existences of a corporate firewall and personal firewalls on every desktop is defense in depth even if it is not typically thought of in this way. So much the better, even if this is defense in depth by accident. (How to tell if it is an accident? Compare the configuration parameters of all the desktop firewalls with each other and with the corporate perimeter for logical synchronicity. You will not find a smooth-walled and narrowing funnel but instead a maze of twisty passages, all subtly different.)

So what would defense in depth look like? Here’s where, for once, we actually get lucky. As we said above, when your primary defenses (firewalls, antivirus, intrusion detection, patch management, ad infinitum) fail, there are only two things that can happen that matter to any appreciable degree: Either some bit of data will be altered (and we are including both data in the colloquial sense and data in the sense of program files), or some bit of data will be sent to a distant location that is unwanted (embarrassing, dangerous, expensive, ...). We are thus made entirely lucky in that an effective defense in depth strategy has a coverage requirement for exactly two things: Don’t let those bits get modified if they should not be modified, and don’t let them fly away if they should be caged. On the assumption – a really BIG assumption but also a very credible one – that your computing environment has enough structure that you know what bits are what, then symptomatic relief of the coughing (unwelcome bit twiddling) and sneezing (unwelcome bit discharge) is going to be quite good enough. How then might we do this?

In other words, the Reference Monitor is an idea whose time has come.

5. The Wisdom of Our Elders

Without waxing rapturous about ancient stone tablets, what we have to do is resurrect a very good idea from a couple of decades ago. The reason we have to do this is simple: the ideas back then were theoretic but today they are practical. Then, data (excepting the nuclear launch codes) wasn't valuable enough, data wasn't exposed and in motion enough, and Moore's Law had not had time to work its compound-interest magic on how much horsepower we could throw at what. We speak, of course, of the Reference Monitor. Since it was first proposed in 1972 and codified in 1983, a lot has changed – the creation of the Internet, the desktop computer, cheap storage, and cheap cycles just to pick a few. In other words, the Reference Monitor is an idea whose time has come.

So, what is a Reference Monitor and why has its time come? By analogy, a reference monitor is to an operating system as a conscience is to an honest person. It watches what other processes do and, where necessary, intervenes; otherwise it is, like the very best security products, entirely invisible and entirely inescapable. A good conscience is like that, too; just as no one wants to live with people who do not have a conscience it is now time to say that no computer that has its hands on valuable bits should not have a reference monitor.

The reference monitor as originally proposed was all about classified data in the military sense. When a document was created, its maker would attach an indelible label that said, for example, "Top Secret." From that time on, any computer handling that document would have a reference monitor which would ensure that the Top Secret label was obeyed by whatever processes fiddled with it. This meant that those processes did not have to be totally hardened and fail safe, just the reference monitor had to be. This is efficient and a much better guarantee, just like it is better to have a professional police force than it is to have random vigilantes on alternate Thursdays.

The reference monitor as proposed then assumed a military grade computer operating system. That never happened. The reference monitor being proposed today (by Verdasys) is more than good enough for commercial purposes even if it is not the answer for protecting the nuclear launch codes. The difference? Data labeling in the sense originally meant – we suggest that indelibly labeling everything that is not part of the furniture at the moment of its creation is infeasible in a world where profit and loss matter. Instead, we remind ourselves that the function of the data label is to make it possible for the reference monitor to be a servant of those labels and not a decision maker about whether this or that content is good, bad, top secret, or whatever. If the reference monitor needs labels to operate but it is commercially infeasible to do labeling, what can we do? We can infer label from context. If we can say "This is the HR database, that is the source tree, and this is \\WINDOWS\system32" then we can say something meaningful over what it is that the reference monitor should do. Put differently, we are accepting existing structure as meaningful and intentional and are thus enforcing policy based on the that meaningful and intentional structure.

A Reference Monitor should be complete, isolated and verifiable.

“Policy?” we hear you say. Yes, policy – all programmatic security is about implementing policy. For the reference monitor to know when to intervene, when to make a log entry, or when to stand silent there has to be a logic to execute. Sure, you could bake that policy into the reference monitor and that, to an extent, is what all the wishful thinking about protecting media files in arbitrary locations (Movie X on Laptop Y in Country Z) is all about – though you have to either get everyone on the planet to implement a reference monitor against the wishes of the user base or you have to attach the reference monitor to the mediafile itself. For those pursuing this dream, we wish them well and we are certain that they are kind to their mothers.

No, the issue is versatility which requires the ability to adapt. As with all of life, too much adaptation is as bad as not enough; the trick is getting the adaptability just right so that the ability to adapt is a net benefit to the honest side of the ledger and not just another acre of attack surface for the dishonest side. Reminding ourselves that we are trying to solve real problems in the real world, we need to have knobs to adjust when conditions change. In other words, the inputs to our solution are these: valuable data that has inferable characteristics (i.e., is not sprayed at random all over creation), a reference monitor that fulfills the tests for a reference monitor to be a reference monitor, and a language in which you can express what it is that you want rather than having to enumerate all the details of how.

Valuable data? Got that. Valuable data with enough structure to make some decisions? Got that. A reference monitor that fulfills the tests for a reference monitor to be a reference monitor? Do we have that? Well, here are the rules – all three of them (and this is a quoted passage from the definitive text):

A Reference Monitor should be:

- a. complete, i.e., it mediates every access,
- b. isolated, i.e., it cannot be modified by other system entities, and
- c. verifiable, i.e., small enough to be subjected to analysis and tests to ensure that it is correct.

That we have, too. The name of this thing is “Digital Guardian” and it mediates every access to data: At the high level this means every operation on data in any of the file subsystem, the clipboard, the network, CD burning, or printing. That every access is mediated is both wonderful and terrible; wonderful in that this is how you pass test (a) and terrible in that the number of recorded events is stunning (1,450 before Microsoft Word can light up its first pixel). Suffice it to say that just like nuclear physics, you want your cloud chamber to catch every particle but you don’t want it to report out the existence of the electron. In other words, catch everything so that you get completeness but then do serious data reduction so that instead of 1,450 system-level events you get one logical event: John opened Doe.doc.

The fact that users of this product continue to surprise its makers with the adaptive re-use of the product's functionality is a head scratching pleasure.

As to isolated, the answer is to a degree self referential. If you have a process running at highest priority as close to the iron as is possible, then it can protect itself in a lot of ways not the least of which is forbidding its own files to be read much less modified, and at the very least ensuring that overwhelming forces (such as shutdown, pull the drive, image that drive on a different OS, and restore) can not and will not produce a record that obscures the logging discontinuity that happened. That is a defense in depth strategy within just this product – what you cannot prevent you must not allow to be invisible. So, yes, this is a reference monitor that passes the test of whether it is isolated or not.

The last test, is it verifiable, is the hardest one for all the usual reasons – telling whether a piece of code does exactly what you want it to do and nothing more is actually impossible at anything more complex than “ADD X Y.” Given the impossibility of absolute proof the preferred alternative is to get outside evaluations of the product from several angles by competent but non-communicating experts. If they are any good, what they will give you is not a statement like “System X cannot be broken” but rather “Breaking System X requires two man-weeks of labor by a world-class team with the following tools.” The risk management decision maker then makes a decision based on whether that level of effort is or is not plausible to worry about given the value of the data at risk, the number of enemies the firm has, and the impact of an undiscovered breach. (Computer scientists will recognize this as yet another form of the “halting problem.”)

The other point about versatility is that the corporate risk manager will want to get “the big picture,” viz., to see all the risks the firm faces and to rationally allocate monies to mitigations. To do this requires integration of data sources that are collected by different instruments, what is called data fusion in the trade. For security products, the true cost is and always has been the cost of integration and a reference monitor (and the records of events seen and actions taken that it produces) is no different -- the true cost is the cost of integration. The product in question, Verdasys' Digital Guardian, is built with this integrability in mind:

- rule language in machinable XML
- code base in localizable UniCode
- database schema published and, excepting for housekeeping issues, database independent
- HTTP or SSL transport of data from collection points to aggregation points
- platform dependencies isolated to the collection side only
- tolerant and adaptive to intermittent connectivity throughout
- export of data either raw (massive) or processed down (in any sense)

This, of course, means that the product can be used in a versatile manner including in ways that may have little to do with security, e.g., inventorying the pattern of application usage so as to calibrate spending on licenses, ascertaining who has not retrieved a globally distributed document that all should see, forbidding data deletion within the scope of a newly served subpoena, and so forth and so on. The fact that users of this product continue to surprise its makers with the adaptive re-use of the product's functionality is a head scratching pleasure.

6. How Do You Know This Works?

How do we know that this works? Bingo...the right question. The security market is full of nothing so much as overwrought solutions for small but well publicized problems, and wishful thinking. Let us first restate the claim and then offer the proof. The claim is this: The failure of any conventional primary protection system (anti-virus, firewalls, patch management, intrusion detection and/or prevention) is unmitigated in that none of those systems cross-protect each other and thus there is no defense in depth. When they fail, they share a common downside risk: data (broadly defined) may be modified or data may be redistributed. If there is to be any defense in depth, it has to be ready to protect data even when the primary protection systems have failed including when they have failed silently. Of course, a hostile insider with motive and opportunity is precisely equivalent to the failure of any or all of the primary protection systems, so a defense in depth strategy that is actually meaningful will make no distinction between the protections afforded against a hostile (or inept) insider versus a hostile outsider (if for no other logical reason that the first measure of success of an outside attacker is to gain the credentials of an insider thus reinforcing that a defense in depth approach to the hostile insider will largely solve the outside attacker problem as a side effect).

To test this (as yet) theory, one would take a set of otherwise comparable computers and, like a good scientist, perform reproducible experiments like this: For a machine that is just as it comes from the manufacturer, for a machine that is as above but which has been locked down in a competent fashion, for a machine that has been locked down plus has had a full array of the conventional primary protection systems installed, and for a machine that is locked down with a full array of conventional tools plus a reference monitor you would then expose them all to a series of valid and plausible attacks of increasing virulence until, one by one, they succumbed. This should remind you of Olympic high-jumping; you keep making the jump harder until only one jumper can clear and then declare success, only in this case you would keep raising the bar until even the last jumper (system) failed. This would include recently crafted malware that is both loose in the wild and the skills and authorities of someone with top-level system privilege. The former will tell you whether the outside attacker can gain the credentials of the insider by any currently known means while the latter, in the truest sense of defense in depth, effectively stipulates that an attacker has in fact gained the credentials of a trusted insider.

If such a test is to be successful, it needs to return a level of effort that is substantially in excess of the plausible risk limit that the firm is willing to tolerate, i.e., the result has to be that, say, only a national laboratory of a hostile power could break in and even then there would be a dying gasp of the computer in question that could not be silenced. Construction of such a test is a current need and we invite parties to participate with us in deciding what is a convincing level of proof from their point of view. Collaboration of this sort is a gating necessity for a valuable result.

So a defense in depth strategy that is actually meaningful will make no distinction between the protections afforded against a hostile (or inept) insider versus a hostile outsider.

Only in field demonstration will the last word be found, but that is silently underway at real customers with real needs.

7. Recap and Challenge

We have talked about a nasty world, what our strategies and choices are for dealing with a nasty world, how we can make progress by borrowing from other fields, why we should now remember the wisdom of our elders, and why versatility is the key to winning an(y) arms race. To some degree, we have asked you to read on faith, but to a greater degree all we have asked is for you to follow our train of logic. We have made the point that conventional primary protection systems are mutually redundant yet are not collectively exhaustive, that they leave holes which can only be addressed by a defense in depth strategy that is not just a marketing doodad. We have (more than) suggested that if ever there was a place where working smarter made more sense than working harder, then digital security is that place and in that spirit we have offered a modern rendition of a venerable truth: a solution to this set of equations, the optimal solution in practical fact. We spoke of general truths for as far into this paper as we could, and then we moved into the particular innovations which we wanted and want you to accept as truth and progress. Only in field demonstration will the last word be found, but that is silently underway at real customers with real needs. We would like to work with you, too.

ABOUT VERDASYS

Verdasys is a developer of software solutions that provide “complete protection” for intellectual property and other mission critical data vital to operating the modern global enterprise. These solutions assure that information accessed and shared within the organization, and with its partners and customers, is secure and handled according to corporate policy and government regulations governing its use. Verdasys is a pioneer and recognized leader in “point of use” data security technology, and is considered a critical technology provider to recognized leaders in insurance, finance, banking, automotive, manufacturing, high tech, and other industries worldwide.

VERDASYS.

950 WINTER STREET, SUITE 2600 WALTHAM, MA 02451
781-788-8180 TEL 781-788-8188 FAX

INFO@VERDASYS.COM WWW.VERDASYS.COM